REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	11.Jan.99	THESIS

4. TITLE AND SUBTITLE
VISUAL COMPUTER SIMULATION OF MATERIAL MICROSTRUCTURE CHANGES DUE TO MANUFACTURING PROCESSES

5. FUNDING NUMBERS

6. AUTHOR(S)
2D LT MORRIS LANNY J

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
LOUISIANA TECH UNIVERSITY

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
THE DEPARTMENT OF THE AIR FORCE
AFIT/CIA, BLDG 125
2950 P STREET
WPAFB OH 45433

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
FY99-61

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Unlimited distribution
In Accordance With AFI 35-205/AFIT Sup 1

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

14. SUBJECT TERMS	15. NUMBER OF PAGES
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

VISUAL COMPUTER SIMULATION OF MICROSTRUCTURE CHANGES IN METALS DUE TO MANUFACTURING PROCESSES

by

Lanny Jason Morris, 2LT, USAF

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science in Mechanical Engineering

COLLEGE OF ENGINEERING AND SCIENCE
LOUISIANA TECH UNIVERSITY

November 1998

LOUISIANA TECH UNIVERSITY

THE GRADUATE SCHOOL

October 19, 1998
Date

We hereby recommend that the thesis prepared under our supervision by Lanny Jason Morris entitled Visual Computer Simulation of Microstructure Changes in Metals Due to Manufacturing Processes

be accepted in partial fulfillment of the requirements for the Degree of Master of Science in Mechanical Engineering

Supervisor of Thesis Research

Head of Department

Mechanical Engineering
Department

Recommendation concurred in:

Advisory Committee

Approved:
Director of Graduate Studies

Approved:
Director of the Graduate School

Dean of the College

GS Form 13
2/97

ABSTRACT

This thesis focuses on the design, algorithm development, and demonstration of a computer program that visually simulates the results of manufacturing processes on the microstructure of metals. The visual simulation program presents an image of grain structure similar to images of etched grains visible in a microscope. The program simulates the response of single-phase pure metals that undergo the processes of grain deformation, grain recrystallization, or grain growth. The computer program is designed to be integrated into any Windows 95/NT program that requires simulation of the microstructure of metals. Object-oriented methods are used in the design of the program to separate the information into objects that the computer can manipulate. Microstructure simulations generated by the program are compared with microstructures generated by other proven programs and with microstructures found in the literature. The results show that a computer program generates simulations of microstructures to the computer screen that visually approximate real microstructures and are equivalent to microstructures generated by other programs. Conclusions are drawn and possible future work is outlined.

APPROVAL FOR SCHOLARLY DISSEMINATION

The author grants to the Prescott Memorial Library of Louisiana Tech University the right to reproduce, by appropriate methods, upon request, any or all portions of this Thesis. It is understood that "proper request" consists of the agreement, on the part of the requesting party, that said reproduction is for his personal use and that subsequent reproduction will not occur without written approval of the author of this Thesis. Further, any portions of the Thesis used in books, papers, and other works must be appropriately referenced to this Thesis.

Finally, the author of this Thesis reserves the right to publish freely, in the literature, at any time, any or all portions of this Thesis.

Author 

Date 23 OCT 98

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

There exists a need in the materials processing field to design optimal manufacturing processes. Engineers in the Materials and Manufacturing Directorate of the Air Force Research Laboratory at Wright-Patterson Air Force Base, Ohio are developing a simulation system for the design of multi-stage material processes. One aspect of this simulation is the optimization of microstructure development. The engineers at the Air Force Research Laboratory are developing methods for simulating the microstructure development; however, they do not have a method to visually display the resulting microstructure to the user (Malas and Frazier, 1997; Malas et al., 1997). As a means of closing a gap in this technical capability, this thesis focuses on the development of software to visualize microstructure in single-phase metals.

The software developed for microstructure visualization through this thesis has several benefits. The first benefit is the ability to show the microstructural development of a simulated metal to the program's user as if the user were performing a "real" manufacturing process and looking at the "real" resulting microstructure under a microscope. A secondary benefit exists in that this program is developed as a separate module of a larger Air Force Research Laboratory simulation system which permits simulation of manufacturing processes. Although this program cannot perform the

calculations to determine microstructure characteristics based on manufacturing processes, it can show microstructures based on input of characteristic data from the Air Force Research Laboratory Simulation or any other source of microstructure characteristics. The program can also show microstructure changes as a result of changes to the metal's characteristics. This ability benefits any engineer working in the metals area, including manufacturing engineers and instructors in materials and manufacturing.

1.2 Literature Review

Examining the effects of metals processing on microstructure evolution is important because the microstructure of a metal effects the strength and physical behavior of the metal. Before computers, the only way to view the changes in microstructure based on metals processing was to process the metal and then examine it. This procedure was expensive in time, money, and material. With computers, however, metals processing and microstructures can be simulated, reducing the cost of testing the actual metal.

The following literature review covers published material about the effects of metals processing on microstructure evolution and the computer modeling of microstructure. The first section reviews literature on the topic of the effects of metals processing on microstructural evolution, and the second section covers the topic of computer simulation of microstructures.

1.2.1 Effects of Metals Processing on Microstructure Evolution

Before microstructural evolution can be simulated, a basic knowledge of how metals processing effects the microstructure must be attained. This section reviews

literature on the topic of microstructural evolution. First, a basic description of the general characteristics of microstructures is presented. In order to describe the microstructure, it is important to have a basic understanding of its characteristics. The next subsection describes the various mechanisms that occur during metals processing that change the characteristics of the microstructure. The final subsection covers the hot rolling process. The hot rolling process is a good example of how microstructural evolution mechanisms occur during metals processing to change the microstructure. This section covers the background information needed to be understood before the computer simulation can be performed.

1.2.1.1 Characteristics of Microstructures. A review of the characteristics of microstructures is necessary before the mechanisms that act on those characteristics can be covered. This subsection covers those characteristics starting with a description of grain size. The topic of grain boundaries is covered next. Finally, grain orientation is described.

1.2.1.1.1 Grain Size. One characteristic that defines a microstructure of a metal is grain size. As individual grains are extremely small and numerous, it is impractical to measure individual grains. Several procedures have been devised to determine an average grain size for a metal. Three of these methods are the Comparison Procedure, the Planimetric or Jeffries' Procedure, and the Intercept Procedure. As these and other methods produce representations of grain sizes in different units, ASTM has devised a standard grain size number representation.

The comparison procedure is used on completely recrystallized or cast metals composed of equiaxed grains. In this procedure, a photograph of the microstructure is directly compared with photographs of standard specimens of known grain sizes. The grain size associated with the standard specimen that most closely matches the test specimen is reported as the grain size for that test specimen (ASTM, 1989).

In the planimetric or Jeffries' procedure, a circle or rectangle of known area (usually 5000 mm^2) is drawn on a micrograph so that at least 50 grains can be counted. The number of grains within this circle (or rectangle) is counted with each grain intersected by the outer perimeter counted as a half-grain. The resulting number of equivalent grains is multiplied by the Jeffries multiplier corresponding to the magnification of the specimen. The result is given in units of grains per square millimeter (ASTM, 1989).

The various intercept methods are generally more convenient to determine than the planimetric procedure. The intercept methods are also good for determining grain size of non-equiaxed grains. The general procedure involves counting the number of grains along a line of specified length so that at least 50 grains are counted. The length of the test specimen, the number of grains intercepted, and the magnification of the sample are then cross referenced with a chart to determine the corresponding ASTM grain size number. For the intercept method, the test length can be a straight line, a circle, or a polygon (ASTM, 1989).

The grain size of a microstructure represents an estimate of size based on these various measurement techniques. The ASTM grain size number represents grain size, as it is independent of the system used to estimate grain size. Several equations have been

developed to convert between the results of the various measurement methods and the ASTM grain size number (ASTM, 1989).

Based on the various measurement techniques, grain size can be easily determined from a micrograph. A standard ASTM grain size number can represent the grain size. Grain size is a measurable standard characteristic that defines a microstructure.

1.2.1.1.2 Grain Boundary. Grain boundaries are important features that describe a microstructure. In a general sense, grain boundaries are simply the locations where grains meet. Analytically, grain boundaries are described as a separate thin phase, which contains a certain quantity of free energy. Grain boundaries act as nucleation sites for new grains during recrystallization.

One manner in which grain boundaries are described is as a phase separate from the primary phase. This boundary phase has a quantity of free energy associated with it in the same way that other phases have an associated quantity of free energy (Fredriksson, 1990). In comparison with the other phases of the microstructure, the grain boundary phase is a high-energy region (Porter and Easterling, 1992). The grain boundary acts as a transfer region for energy between the separate grains.

The grain boundaries are important in determining rate of grain growth, as well as recrystallization, because they act as nucleation sites for new grain formation. In grain growth, energy is transferred so as to decrease the amount of energy in the grain boundary. This process results in grain growth as the grain boundary migrates away from the growing grain. In the process of recrystallization, atoms are transferred from the cold-worked original grain across the grain boundary into a new recrystallized grain. This process continues until an equilibrium state is reached (Fredriksson, 1990).

The grain boundaries of microstructures are treated as regions of high energy that form the borders between grains. These areas of high energy act as interfaces for the transfer of energy between grains. The resulting energy transfers cause grain growth and recrystallization.

1.2.1.1.3 Grain Orientation. The orientation of grains in a microstructure is an important property that can effect grain growth and recrystallization. There are three different types of orientation systems. The three orientation systems are called completely oriented, partially oriented, and oriented planer surfaces. In expressing grain orientation, grain elongation is an important factor. Grain orientation affects the surface energy along the grain boundary.

One type of orientation that occurs in microstructure is a system of lines parallel to each other. These lines have clear, recognizable orientation directions. If the microstructure contains nothing but parallel linear elements, the microstructure is considered a completely oriented system. A completely oriented microstructure can contain several oriented elements with different orientation directions (American Society for Metals, 1973).

The partially oriented system is another type of orientation that can exist in a microstructure. This system consists of a combination of oriented and random elements. The oriented elements are parallel and linear in specific directions similar to a completely oriented structure. The random elements are mixed in with these oriented elements (American Society for Metals, 1973).

The third type of oriented system consists of oriented planar surfaces. Examples of this type of oriented system include pearlite in steel, lamellae in unidirectionally

solidified eutectics, and lamelar precipitates. The planar surfaces are oriented along definitive orientation planes (American Society for Metals, 1973).

In some of these orientation systems, grain elongation is an important property. A shape index is a good method of representing elongation. The simplest index to use is the ratio of mean length to mean width (American Society for Metals, 1973).

Grain orientation causes a variation in the surface free energy of a grain. This variation in the surface free energy causes a variation in the growth rate of the grain. The grain orientation is represented by an anisotropy factor in equations representing grain boundary energy (Fredriksson, 1990).

Grain orientation is an important property in microstructures. Completely oriented systems, partially oriented systems, and oriented planar systems are three different kinds of orientation systems that can exist in microstructures. Grain elongation is a property of the orientation and can be represented by an index. Grain orientation is an important factor in the determination of the free energy in the surface of a grain.

1.2.1.2 Mechanisms of Microstructure Evolution During Metals Processing. Various mechanisms occur during metals processing that change the characteristics of a microstructure. Grain growth is a mechanism that occurs to reduce the grain boundary energy. This mechanism occurs whenever a metal is heated to a certain temperature. Grain recrystallization occurs to reduce the stored strain energy caused by deformation of the metal. Both of these mechanisms can be present during manufacturing processes.

1.2.1.2.1 Grain Growth. Grain growth is one of the major mechanisms that occur in the evolution of microstructures. The driving force behind grain growth is a decrease in

grain boundary energy (Fredriksson, 1990). The mechanism of grain growth seems to follow certain observed "laws," where the term "law" connotes consistently observed behavior rather than a mathematical or deductive proof. One consistent observation is that grain growth in pure metals is different than grain growth in metal alloys.

Grain growth occurs when a metal is heated to a temperature greater than half of its melting temperature. During grain growth, grains with less than six grain boundaries will shrink, while grains with greater than six grain boundaries will grow. This process causes a reduction in the number of grains, resulting in an increase in the average grain size. The final result is a decrease in grain boundary energy (Porter and Easterling, 1992).

The process of grain growth has been observed to obey several "laws." One observation is that grain boundary migration and not coalescence of neighboring grains is the method by which grain growth occurs. Another observation is that the rate and direction of boundary migration are not constant in subsequent heating periods. Grains may be consumed by a neighboring grain while it is growing into another neighbor. As a grain begins to disappear, its neighbors consume it more rapidly. If a grain boundary is curved, the boundary migrates towards its center of curvature. A final observation made is that if the angles formed where grain boundaries meet are not 120 degrees, then the grain within the more acute angle is consumed so that all angles will become 120 degrees (Burke and Turnbull, 1952).

Grain growth can be affected by alloying elements in metals. Alloying elements tend to concentrate in grain boundaries and cause a "solute drag effect." This drag effect causes grain growth in alloys to be slower than in pure metals (Fredriksson, 1990).

Grain growth as a mechanism in grain evolution is driven by a decrease in grain boundary energy. This mechanism tends to obey "laws" that have been observed in experiments. Grain growth tends to be slowed by the effects of alloying elements in a metal. All of these characteristics of grain growth must be considered in any computer simulation of this mechanism.

1.2.1.2.2 Grain Recrystallization. An important mechanism in grain evolution is grain recrystallization. A decrease in the stored strain energy is the driving force behind the recrystallization mechanism (Fredriksson, 1990). A few "laws" have been established to describe recrystallization. In most cases, recrystallization occurs on grain boundaries.

In recrystallization, new grains are formed in order to decrease the stored strain energy caused by cold-working the metal. Recrystallization occurs when the metal is heated to an activation temperature and held at that temperature for a length of time. Nucleation and growth rates describe the recrystallization reaction. Nucleation and growth rates are different from one area to another area and with transformation time (Furu et al., 1990).

Recrystallization tends to follow certain "laws" than have been established qualitatively. In order for recrystallization to occur, a minimum deformation in the metal is necessary. The temperature required for recrystallization increases as the degree of deformation decreases and decreases as the annealing time increases. The degree of deformation is the primary factor in determining the final grain size, with the annealing temperature also affecting grain size to a lesser degree. Generally, the final grain size decreases as the degree of deformation increases and the annealing temperature decreases. Larger original grain sizes require more cold deformation in order to get an

equivalent recrystallization time and temperature. As the temperature of working increases, the amount of cold work must increase to give an equivalent amount of deformational hardening. The final established law is that grain size increases if heating is continued after recrystallization is complete (Burke and Turnbull, 1952).

Generally, recrystallization occurs along grain boundaries. New grains begin to nucleate along grain boundaries until they fill the initial boundaries. This process causes a shell of new grains to form. Once the thickness of the shell reaches the stable dynamic grain size, the shell stops growing. New nucleation sites then emerge along the new grain boundaries, and the process repeats itself. This continues to occur until the original grain is consumed (Sakai and Jonas, 1984).

Recrystallization is a mechanism of grain evolution driven by a decrease in stored strain energy. Several "laws" have been established to describe the mechanism of recrystallization. Recrystallization generally occurs along grain boundaries. In any method to model recrystallization, these observations made about recrystallization need to be considered.

1.2.1.3 Hot Rolling Process Effect on Microstructural Evolution. Hot rolling is a manufacturing process in which a metal is forced through rollers after heat has been applied. The rollers reduce the cross-sectional area of the metal, and the heat allows the grains of the metal to recrystallize during the process. Static, dynamic, and metadynamic recrystallization as well as grain growth can occur during a hot rolling process. Temperature, time, strain rate, strain, and stress on the metal determine how these different mechanisms are going occur in the microstructure (Devadas et al., 1991a).

Dynamic recrystallization occurs while the metal is being deformed in the hot rolling process. Hot deformation causes the dislocation substructure to become dense and inhomogeneous, allowing new recrystallized grains to nucleate. At low strain rates, the bulging of existing grain boundaries forms the nucleating grains. At high strain rates, tangled dislocations developing within the grains form a cellular substructure where recrystallized grains nucleate (Devadas et al., 1991b). In some metals, such as ferritic steels, a process called dynamic recovery occurs at the same time as dynamic recrystallization. Dynamic recovery acts to "undo" the effects of dynamic recrystallization. For these metals, dynamic recrystallization does not occur (Roberts, 1986).

Static recrystallization occurs after hot deformation (Devadas et al., 1991b). Static recrystallization also requires the critical strain to not be exceeded. Like dynamic recrystallization, pre-existing grain boundaries form the nucleation sites. Static recrystallization differs from dynamic recrystallization due to grain growth that occurs during the recrystallization process (Roberts, 1986).

Metadynamic recrystallization occurs during hot deformation if the critical strain is exceeded. This type of recrystallization is a two-stage process. During the first stage, recrystallization occurs in regions not dynamically recrystallized earlier. In the second stage, the residual dislocation density in the dynamically formed grains causes a reaction in the remaining areas. Generally, stage I proceeds faster than stage II (Roberts, 1986).

Grain growth occurs at the end of the process as the metal cools. Grain growth is generally modeled as a parabolic curve that decreases with time. Two explanations for the decrease in grain growth over time have been postulated. One of the explanations

states that the new, recrystallized grains do not have an equilibrium shape, but as these grains grow into an equilibrium shape, grain growth slows. The second postulate states that new, recrystallized grains have boundaries that are free from impurities. As the grains grow, the amount of impurities in the grain boundaries increase causing grain growth to decrease (Roberts, 1986). After the metal has cooled enough, grain growth ends and the hot rolling process is complete.

The manufacturing process of hot rolling is a useful process to model. During the process, as the metal is formed between two hot rollers, all of the mechanisms that have been presented occur. Three different forms of recrystallization as well as grain growth occur. All of these mechanisms are dependent on the conditions the metal undergoes.

1.2.2 Computer Modeling of Microstructures

This section discusses methods of representing microstructures so they can be modeled by computer simulation. In modeling microstructures, several properties of the microstructures need to be considered. These properties include grain size distributions, grain orientations, average grain size, and grain boundary geometries (Krakow, 1995). Three methods of modeling microstructures are presented along with a brief description of other methods of representing microstructures. The first method is the use of a triangular lattice map of points with a Monte Carlo scheme to generate a microstructure. The second method is the use of a Voronoi pattern to represent a microstructure. The third method discussed involves using cellular automata models to represent the grain structure and finite element heat flow equations to grow the grain structures. Other

methods, such as the use of atom probes, are also briefly mentioned, for completeness of the discussion.

1.2.2.1 Modeling Microstructures: Lattice Map with Monte Carlo Method. One way of representing a microstructure is by the use of a triangular lattice map (Figure 1-1). Each point on the map is assigned a number between 1 and Q that represents the orientation of the grain at that point. Values of Q must be large enough so that grains of like orientation will not frequently impinge. Typically values of Q = 48 or 64 are chosen. It must be noted that computational speed decreases as Q increases. According to Grest et al. (1986), values of $Q \geq 36$ produce results that are insensitive to Q. Using the same method, Takayama et al.(1996) used a value of Q = 16.

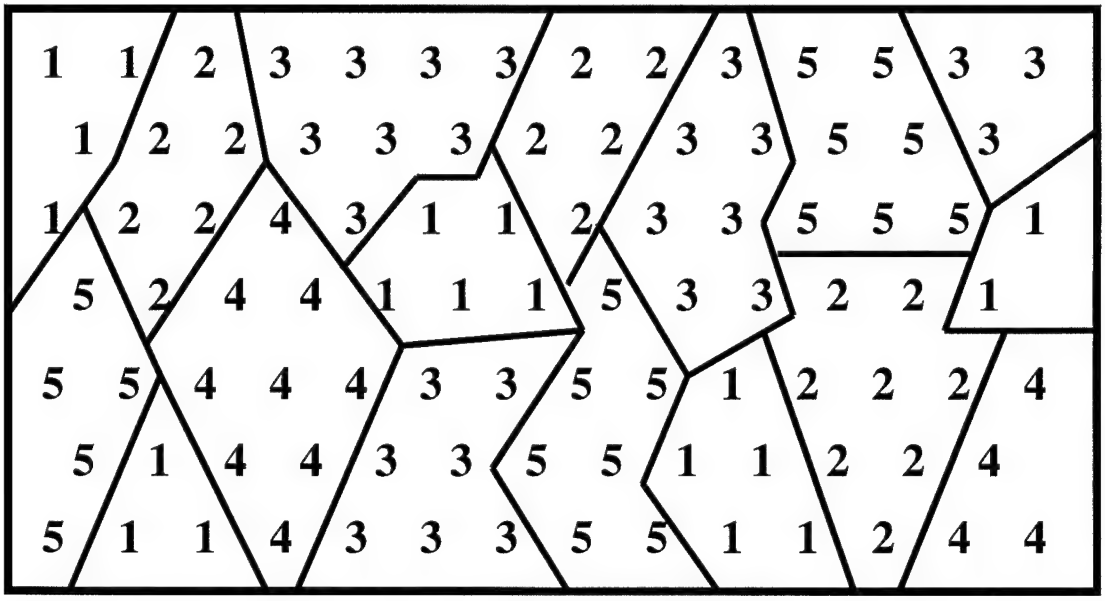


Figure 1-1. Example of the Lattice Map Used With the Monte Carlo Method of Microstructure Representation

The grain boundary in the Grest et al. model is defined to exist between two points of different orientation. The grain boundary energy is defined by the interaction between points on the lattice map within a given distance of each other, usually the

nearest neighbors. The grain boundary energy is defined in Equation 1-1. Where S_i represents the orientation of point i ($1 < S_i < Q$), and $\delta_{A,B}$ represents the Kronecker delta. The sum is taken over all points within a specified distance (Grest et al., 1986).

$$E_i = -J\sum_j \delta_{S_i S_j} - 1 \tag{1-1}$$

Boundary motion is simulated by randomly changing the orientation of a randomly chosen point to a value between 1 and Q. If the change in energy resulting from the re-orientation is less than or equal to zero, then the re-orientation is accepted. For a change in energy greater than zero, the change in orientation is accepted based on a probability of exp (-ΔE/kBT) where kBT is the thermal energy (Grest et al., 1986).

1.2.2.2 Modeling Microstructures: Voronoi Patterns. Another method of representing microstructures is the use of Voronoi patterns. Each grain can be represented by a point. Circles can then be generated based on growth rates of these individual points. Grain boundaries are formed when circles of neighboring grain centers intersect. When the circles of three different grain centers intersect each other, a triple point is formed. This triple point is equidistant to the three-grain centers. The triple point exists at the location where Equation 1-2 is satisfied (Frost and Thompson, 1986). In Equation 1-2, r_n is the range from the triple point to grain n, G is the grain growth rate, and t_n is the time that the grain has existed.

$$r_1 + G \times t_1 = r_2 + G \times t_2 = r_3 + G \times t_3 \tag{1-2}$$

In forming the microstructure by this method, a site is randomly picked as the start of a grain. At a certain time interval, that initial grain grows in size and a new site is randomly chosen for a new grain. If an existing grain already occupies that site, then the

site is skipped. The average grain area, A, is given by Equation 1-3, where N is the nucleation rate per unit area per unit time and G is the growth rate (Frost and Thompson, 1986).

$$A = 1.1371 \frac{G}{N}^{\frac{2}{3}} \tag{1-3}$$

The addition of grains end when the entire sample plane is covered by grains and no more can be added. At this point, the location and nucleation time are known for each grain. The triple point is then found by considering all groups of three grains that are near each other. A triple point is found by determining the location where the first and second hyperbola boundary and the first and third hyperbola boundary meet. The time that this triple point is formed is determined by: $t=(r_1/G)+t_1$. Nearby grains are then examined to determine if another grain would have reached the potential triple point before the initial three. If another grain did reach the point first, then the triple point is not real and is neglected. After all triple points are known, then hyperbola boundary segments are drawn between the triple points to form the grain boundaries (Frost and Thompson, 1986).

Using a simplified version of the Frost and Thompson model, Enomoto et al. (1993) consider the grain boundaries to be straight lines drawn between the triple points. Defining the term vertex to mean triple point, the set of vertices for the microstructure is used in determination of the grain boundary motion.

Enomoto et al. make the assumption that as the total boundary length is reduced, the resulting energy release is dissipated by the frictional force against the vertex motion. The sum of the energies of all the grain boundaries constitutes the total energy of the system. Making some approximations and simplifications, Enomoto et al. find the

friction tensor equation to be $D_{ij} = \eta r_{ij}/6$. In this equation, D_{ij} represents the friction tensor, r_{ij} is the vector between the i and j vertices, and η represents the friction constant between the i and j vertices (the value of η is assumed to be constant for all values of i and j). Using this method, grain boundary migration can be simulated (Enomoto et al., 1993).

1.2.2.3 Modeling Microstructures: Cellular Automata Methods. Rappaz et al. (1995) have proposed the use of cellular automata models to represent microstructural features. These models are based upon the growth of dendrites. Finite element heat flow calculations are used along with the cellular automata models to predict the development of grain structure.

In the cellular automata method, the area of interest is divided into a large number of cells used for mapping and visualization of the grains. At the start of the simulation, all cells are considered to be liquid with a state index of zero. Before the solidification process begins, a set of randomly selected cells is defined as "nucleation cells." The nucleation cells are given nucleation undercooling temperatures. If the uniform undercooling of the specimen becomes larger than the nucleation undercooling temperature of a cell, nucleation can occur in that cell provided it is still a liquid. The state index is then changed to either represent the number of grains (eutectic alloy) or the grain orientation (dendritic specimen). A growth kinetics law is used to update the grain envelope at each time step. Analytical equations can be used for a uniform temperature situation; however, if the temperature is non-uniform, a finite element model can be used (Rappaz et al., 1995).

1.2.2.4 Other Methods Used to Represent Microstructures. Other methods that represent microstructures involve the use of atom probes, field ion micrographs, and field evaporation microscopy. These methods involve obtaining data from sensors or probes and rendering the microstructures on a computer based on the data (Cerezo et al., 1992). The results are similar to a photograph. Since a photograph of a real microstructure is not desired for this approach to virtual manufacturing, these methods are mentioned only for completeness.

1.3 Purpose

The purpose of this thesis is to demonstrate the capability of computer software to generate a visual representation of the microstructure of a metal, given initial data. The software transforms the initial microstructure into a new microstructure based on new data. The new data are obtained from microstructural data gathered in the literature survey. The software is written in Microsoft Visual C++ and is designed for a Windows 95/NT based platform permitting the integration with the Wright Laboratory Simulation. This software system provides new capability for metals visualization not presently available in current virtual manufacturing methodologies.

The primary requirement for this software system is to simulate a microstructure of a single-phase, pure metal using the input data of grain size, strain, and percent of recrystallized grains from any source of data. Verification that the simulation performs as expected is proved by comparing the input data to the characteristics of the simulated microstructure. The simulated microstructure is also compared with microstructures of actual metals in order to validate the results.

1.4 Summary of Remaining Chapters

This paper presents the solution to the problem of designing a computer simulation to visually represent the effects of manufacturing processes on the microstructure of a metal. The next chapter describes how the simulation is used and how the algorithms of the simulation work. Chapter three details the testing and evaluation process and presents test results. The final chapter concludes this thesis by summarizing the work, evaluating the procedure used to solve the problem, and suggesting improvements that can be made to the simulation for future work. A code listing of the C++ program can be found on a diskette accompanying this thesis.

CHAPTER 2

DESCRIPTION OF MICROSTRUCTURE SIMULATION SOFTWARE

2.1 Chapter Introduction

In order to simulate microstructure transformation, four processes are described as algorithms. These algorithms include the initial microstructure algorithm, the grain deformation algorithm, the grain recrystallization algorithm, and the grain growth algorithm. The initial microstructure algorithm generates a Voronoi diagram to simulate the microstructure on the computer's display. The grain deformation algorithm uses the principle of conservation of area and a deformation factor to deform the simulated microstructure. For the recrystallization algorithm, new initial grain centers are generated along the grain boundaries, and a new Voronoi diagram is generated to simulate grain recrystallization. In order to simulate grain growth, triple points are repositioned so that the grain boundaries that form them meet at 120-degree angles.

2.2 Microstructure Representation

Before the simulated microstructure can be created and processed within the computer, the method of representation has to be described. The method that represents the microstructure is composed of two parts. The first important part of the algorithm is the general method used to represent the microstructure, such as the Voronoi diagram

method and the Monte Carlo lattice method. The second part is the method of representing the microstructure as data within the computer.

The Voronoi diagram method and the Monte Carlo lattice method were the two general methods of representing microstructures that were considered. The Voronoi diagram method, shown in Figure 2-1 allows the microstructure to be represented as initial grains centers and vertices that represent the end points of grain boundaries. The Monte Carlo lattice method represents the microstructure as many points on a grid. Each point has an orientation number associated with it. Sets of neighboring points with the same orientation compose a grain. The Monte Carlo lattice method requires that each point be represented in the computer.

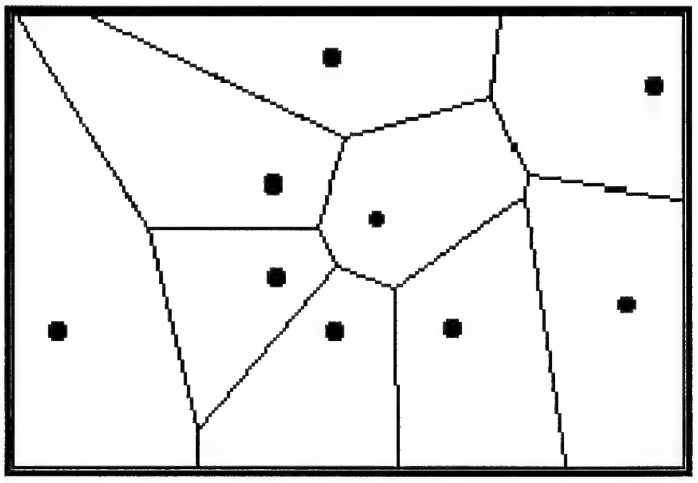


Figure 2-1. Voronoi Diagram.

The Voronoi diagram method of representing microstructures has several advantages over the Monte Carlo lattice method. The Voronoi diagram method requires fewer points to be stored in order to represent a microstructure compared with the Monte Carlo lattice method. Further, the Monte Carlo lattice method uses information about the metal to change the microstructure. Because the program needs to be able to generate general microstructures of any metal that have to be good enough only for human recognition and viewing, the Voronoi diagram method is used in this computer simulation.

Object-oriented methods are used to represent the microstructure as data within the computer. First the microstructure is represented as an object that contains other objects that describe the microstructure and methods that act on those objects. The objects that compose the microstructure object include the grain list object, the triple point list object, and other single variable objects as shown in Figure 2-2.

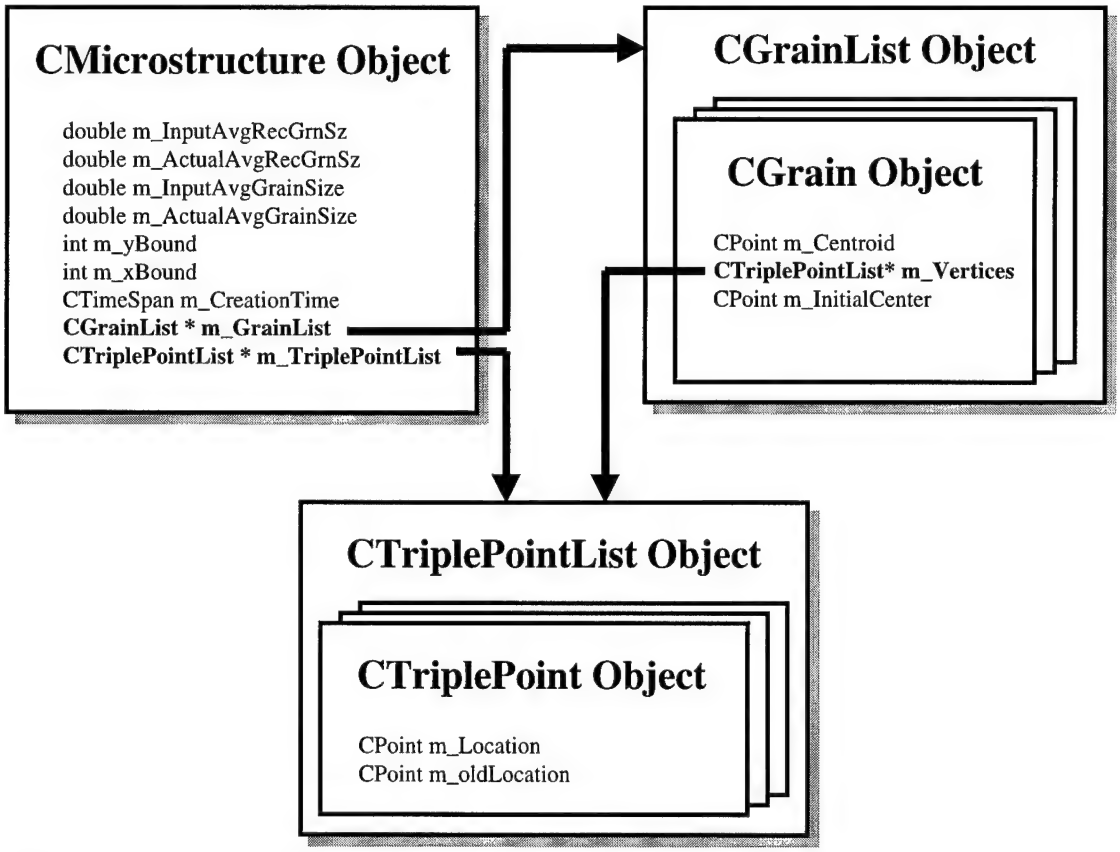


Figure 2-2. Diagram of the Object Relations for the Simulation.

The grain list object is derived from the object list class from the Microsoft Foundations Class (MFC) library. This object contains the individual grain objects that compose the microstructure and the methods used to manipulate and access this list. The various algorithms use the grain list object to access the grain objects either to get information from them or to change them.

The grain objects contain objects that define the individual grains and methods used to manipulate the grain information. Those objects include a triple point list object of vertices and the locations of the initial grain center and the centroid. The methods perform such tasks as calculating the grain area, sorting the vertices, and other operations to an individual grain.

The triple point list object is found in both the microstructure object and the grain object. This object, like the grain list object, is derived from the object list class of the MFC library. Triple points are contained within the triple point list object. In the microstructure object, the triple point list contains all of the triple points generated to form all of the grain boundaries. In the grain object, the triple point list contains the triple points that form the vertices of the grain.

2.3 Initial Microstructure Algorithm

2.3.1 Demonstration Description

After the user starts the program, a window, shown in Figure 2-3, is displayed. This window asks the user to input the desired initial average grain size in micrometers and the number of grains to display for the microstructure. The algorithm automatically generates the random number generator seed based on the system time, so the user can keep the seed number or change it. After the

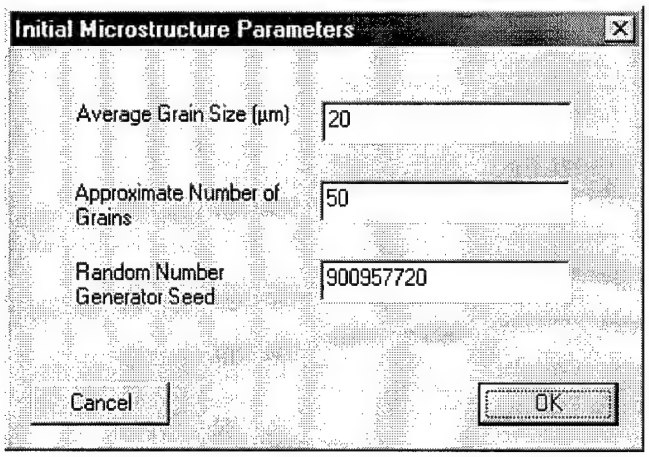


Figure 2-3. Initial Microstructure Parameters

required information has been entered, the program generates the microstructure and displays it in a new window, shown in Figure 2-4. The user can then apply a transformation to the microstructure by choosing one from the "Transform" menu.

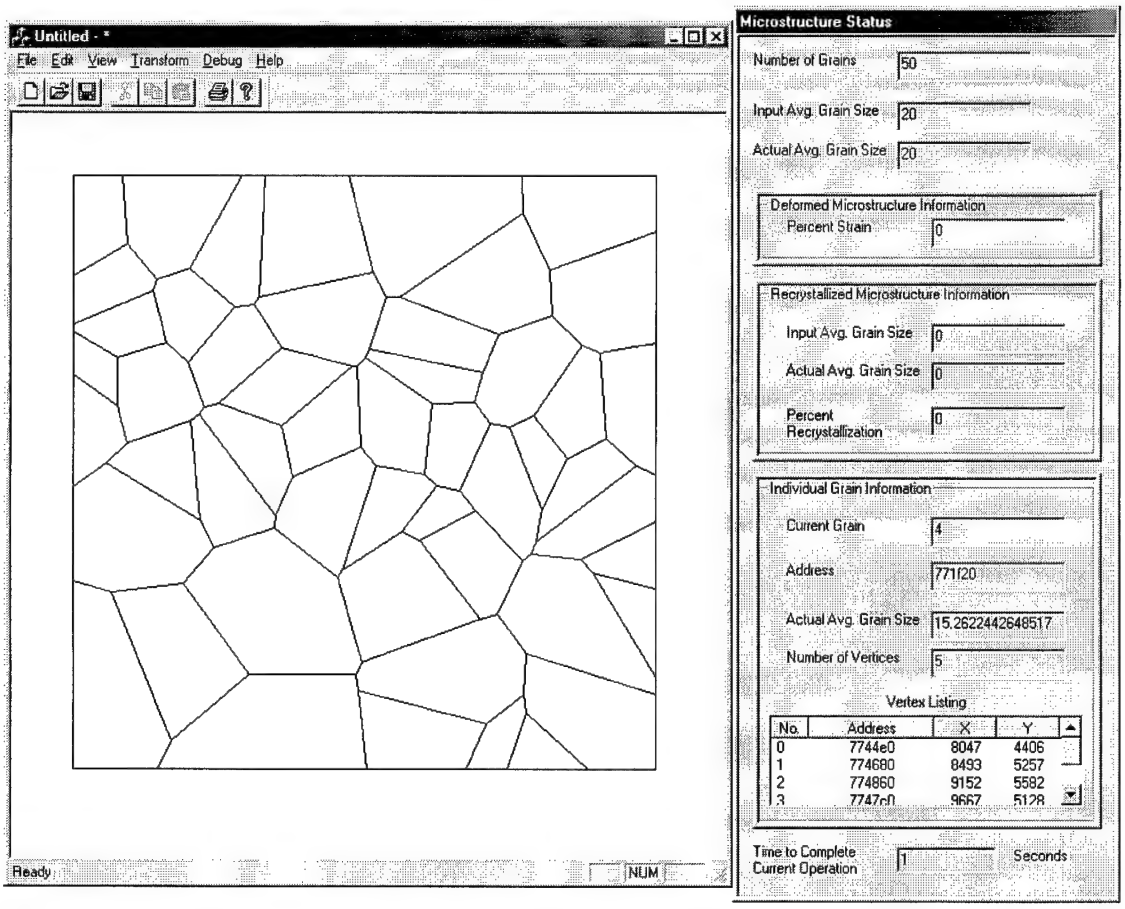


Figure 2-4. Results of Initial Microstructure Algorithm.

2.3.2 Algorithm Description

Now that the method of representing the microstructure has been determined, the method of generating the initial microstructure has to be described. Values for the desired average grain size, scale factor, and random number seed are passed to the algorithm when it is called by the main program. Several steps are involved in the initial microstructure algorithm (shown in Figure 2-5). The first step is the generation of the initial grain centers. Once these points have been generated, they have to be processed

through a Voronoi diagram algorithm to find the triple points. Once the triple points are found, the grain boundaries are clipped so that the Voronoi diagram is contained within a bounding area. After the Voronoi diagram has been bounded, corrections are made to triple points that have more than three grains associated with them. The actual average grain size is then calculated and stored in a data member, and the operation's time is stored in another data member.

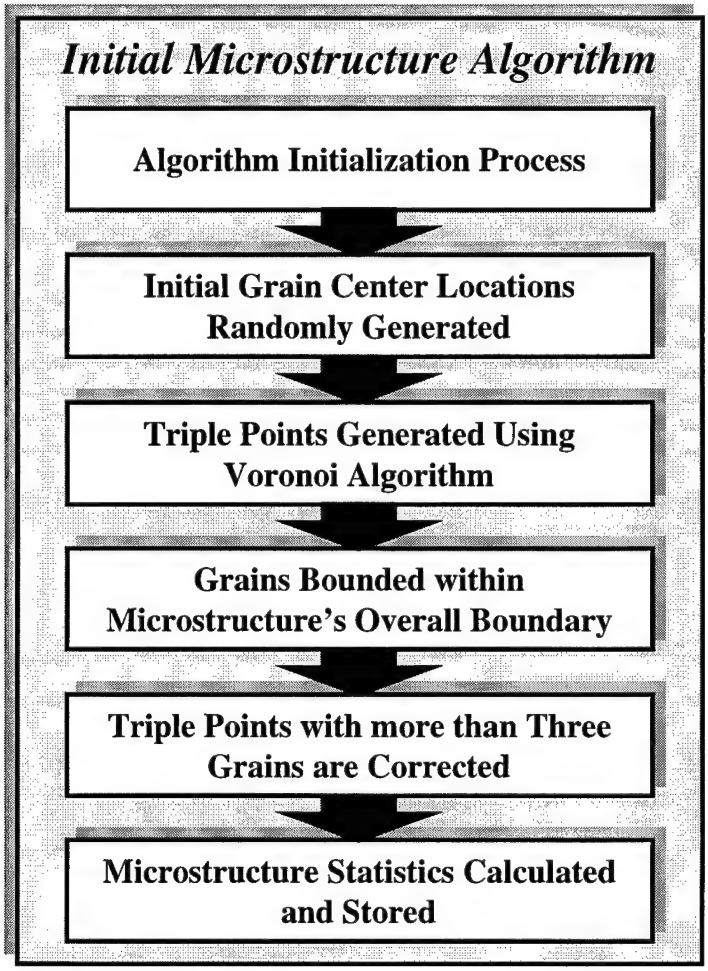


Figure 2-5. Initial Microstructure Algorithm Process Diagram.

Before the initial grain center locations are generated, the x and y boundaries are set based on the scale factor. The microstructure is then cleared, which involves clearing the grain list and triple point list. The number of grains needed is based on the desired grain size and the scale factor. Four boundary grains are created with initial grain centers at (x, y) coordinate positions (-scale factor, -scalefactor), (-scale factor, +2*scale factor), (+2*scale factor, -scale factor), and (+2*scale factor, +2*scale factor) and added to the empty grain list. The triple point finding algorithm needs three grains in order to find the triple point between them. Because of this

requirement, grains on the edge of the bounding area would be missing triple points and therefore have missing grain boundaries without these boundary grains. The standard random number generator is initialized with the random number seed. X and y coordinates are randomly generated and compared with the x and y coordinates of existing grains. If the new coordinates are not the same as an existing set of coordinate, a new grain is created at the new coordinates and added to the grain list. If the new coordinates are the same as an existing set of coordinates, the new coordinates are randomly generated again and rechecked with existing grains.

After the grain list is generated, the triple points are found. In order to find the triple points, three grains are chosen and the point that is equidistant from their grain centers is located based on Equation 2-1, where (x,y) is the triple point location and where (x_i,y_i) is the center of the i'th grain. After the equidistant point is found, the distance between this point and the other points in the grain list are compared with the distance between this point and the three grains of interest. If a grain center other than the three grain centers of interest is closer to the equidistant point, then this equidistant point is not a valid triple point for the Voronoi diagram. If no other grains are closer, then the triple point is considered valid and added to the microstructure's triple point list. If the triple point is not in a vertex list of a grain of interest, then the triple point is added to that grain's vertex list. This procedure continues until all three-grain combinations have been considered. After the triple points have been found, the triple points in each grain's vertex list are sorted in a clockwise order about the grain's initial center.

$$(x-x_1)^2+(y-y_1)^2=(x-x_2)^2+(y-y_2)^2=(x-x_3)^2+(y-y_3)^2 \qquad (2\text{-}1)$$

Once the triple points are found and vertex lists are sorted about their initial grain centers, the microstructure is bounded so that no grain boundaries exist outside the predefined bounding area. Except for the four initial bounding grains, each grain is checked for grain boundaries that exist outside the bounding area. Starting with the first vertex and proceeding through the vertex list to the last vertex, the grain boundary between each vertex and the next vertex are checked to determine which vertex or vertices are outside the bounding area. If the vertices of interest are not outside the bounding area, the boundary needs no correction. If one of the vertices of interest is outside the bounding area and one is inside the bounding area, then a new vertex is created at the intersection of the grain boundary and bounding area. The vertex that lies outside of the bounding area is removed from the vertex list. If both vertices of interest are outside of the bounding area, then both vertices are removed from the vertex list. After all of the vertices have been checked, then the vertices are sorted around the initial grain center again. The last step to bound the grains is to find and bound the corner grains. The grains that contain a vertex on each of two bounding lines that intersect are given a new vertex at the intersection of the two bounding lines.

By definition, an individual triple point can only be associated with three grains. However, due to round-off errors, the algorithm will associate some triple points with more than three grains. After the microstructure is bounded, the triple points have to be checked for this condition. The algorithm loops through the master triple point list. For every triple point, it finds the grains, within the grain list, that contain the triple point in their vertex list. If more than three grains contain the triple point in their vertex lists, those grains are placed in a temporary grain list. While that temporary grain list contains

more than three grains, four grains are selected from the list. One grain is chosen, and a new triple point is created along the line between centroid of the grain opposite to the chosen grain and the triple point of interest at a distance of two units along that line away from the opposite grain's centroid. This new triple point is added to the vertex list of the chosen grain and the grains adjoining the chosen grain. The triple point of interest is removed from the chosen grain's vertex list, and the chosen grain is removed from the temporary grain list. If the number of grains in the list is still greater than three, the process is repeated, otherwise, this triple point has been corrected.

After the triple points have been corrected, the microstructure is complete. The final steps performed by the algorithm are to calculate the actual average grain size and to determine the amount of time used by the algorithm. The area of each grain is calculated using the method presented by Allen Van Gelder (1995). The areas are totaled together and divided by the number of grains to compute the average grain area. The average grain size is the diameter of a circle with the same area as the average grain area. The time is calculated simply by reading a timer object initialized at the beginning of the algorithm. The actual average grain size and time are stored in variables in the microstructure object.

2.4 Grain Deformation Algorithm

2.4.1 Demonstration Description

The grain deformation algorithm is accessed when the user selects "Grain Deformation" from the "Transform" menu in the microstructure display window. After the user selects "Grain Deformation," a window appears as shown in Figure 2-6, which requests the user to enter the amount of compressive strain in percent. In order to enter tensile strain, the user enters a negative percent. For this prototype, strain is only considered in the screen-vertical direction. After the user has entered the required information, the algorithm processes the data and redraws the microstructure to display the resulting changes, shown in Figure 2-7.

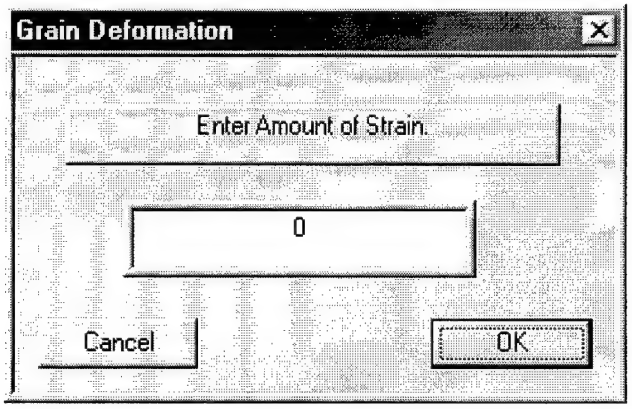


Figure 2-6. Grain Deformation Input Dialog Box.

2.4.2 Algorithm Description

The grain deformation algorithm is a simple procedure, shown in Figure 2-8, that is based on conservation of area, which is the 2-D case derived from the real 3-D case of conservation of volume (Dieter, 1986). This algorithm uses the amount of strain to scale the two dimensional boundaries and to determine an x and y scaling factor. The scaling factors are applied to each of the grains and triple points in the microstructure to adjust

the initial grain centers and vertices. Finally, the actual average grain size and grain centroids are recalculated and stored in data members.

The amount of strain is passed to this algorithm and used to scale the microstructure. The new y boundary is found by multiplying the old y boundary by the

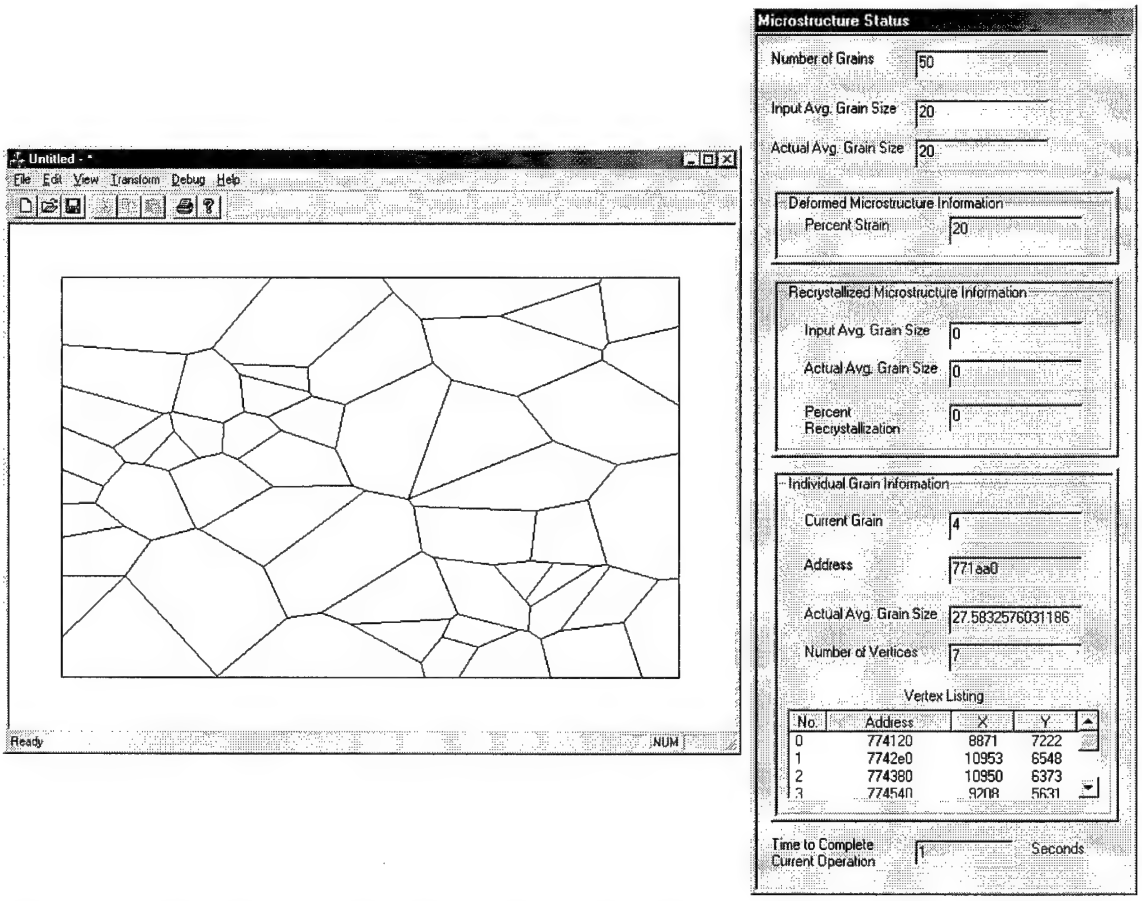


Figure 2-7. Results of Grain Deformation Algorithm.

strain plus one. The new x boundary is found by dividing the new y boundary into the area. The x scaling factor is found by dividing the new x boundary by the old x boundary. The y scaling factor is found by dividing the old y boundary into the new y boundary.

The initial grain centers and triple points are then moved. First, the algorithm moves the initial grain centers by multiplying the current x location by the x scaling

factor and the current y location by the y scaling factor. The algorithm then moves the triple points by multiplying the triple points' x location by the x scaling factor and the y location by the y scaling factor. Once the grain centers and triple points are moved, the microstructure has been deformed.

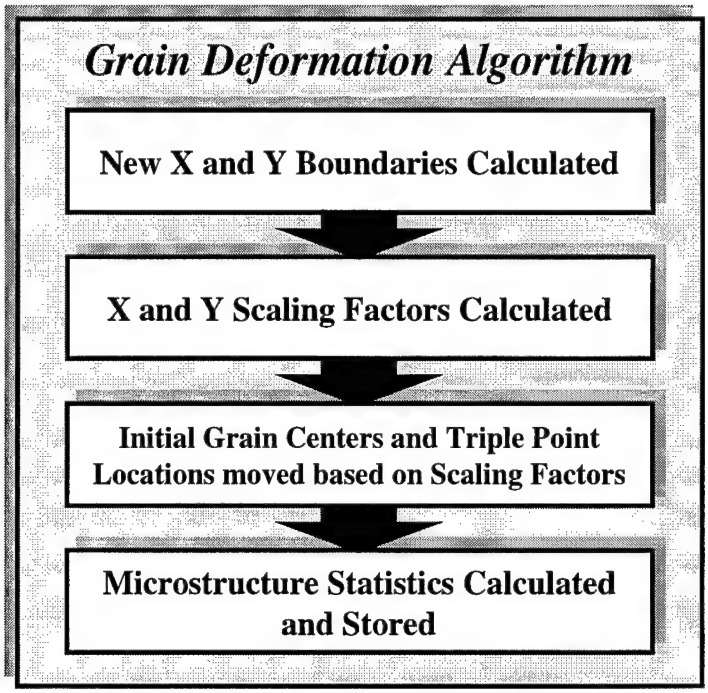


Figure 2-8. Grain Deformation Algorithm Process Diagram.

Before the algorithm is complete, two steps are performed. The actual average grain size is computed again using the same procedure used in the initial microstructure algorithm. This value replaces the old value stored in the microstructure object. The completion time is again calculated and stored in the microstructure object also using the same procedure used in the initial microstructure algorithm. With these steps complete, the microstructure deformation is complete.

2.5 Grain Recrystallization Algorithm

2.5.1 Demonstration Description

By choosing "Grain Recrystallization" from the "Transform" menu, the user initiates the grain recrystallization algorithm. After the algorithm is started, a window, shown in Figure 2-9, appears and asks the user to enter the percent recrystallization and

the new recrystallized grain size, in microns. After the user has entered the data and pressed the "OK" button, the algorithm processes the information and redraws the microstructure to display the results as shown in Figure 2-10.

2.5.2 Algorithm Description

The grain recrystallization algorithm is composed of several procedures, shown in Figure 2-11, in order to simulate the recrystallization process. For this simple model, grain recrystallization is assumed to occur only at grain boundaries. The first procedure is to randomly generate new initial grain centers along the grain boundaries based on the percent recrystallization and the new recrystallized grain size. After the new initial grain centers are generated, new triple points are found. The microstructure is bounded and the triple points are corrected, like the initial microstructure algorithm. Because the triple point algorithm generates new grains with the same average size as the original grains, the vertices of the original grains are scaled in order to correct the size of the new grains based on the percent of recrystallization. The actual recrystallized grain average size, actual average grain size, and algorithm completion time are calculated and stored in data members.

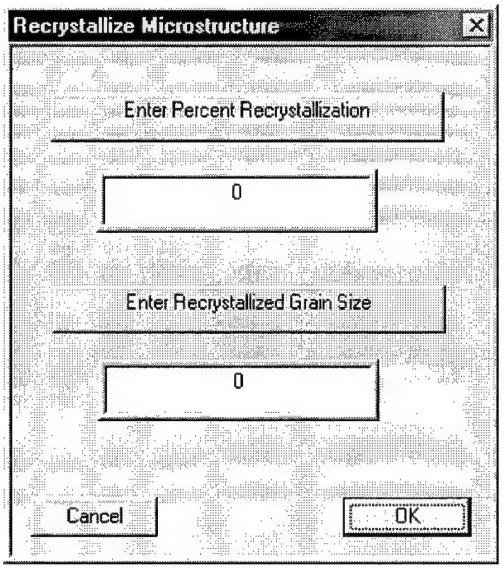


Figure 2-9. Recrystallize Microstructure Dialog Box.

Before new grains are randomly generated along grain boundaries, the grain area for each of the existing grains are calculated and stored in an array, and a temporary grain

list is created to store pointers to these original grains. Also, the initial grain locations are set at the locations of the centroids for all of the original grains except the bounding grains. The total area of recrystallized grains is calculated by multiplying the total area by the percent of recrystallization. The average recrystallized grain area is found by multiplying the square of half the recrystallized grain size by π. The total area of

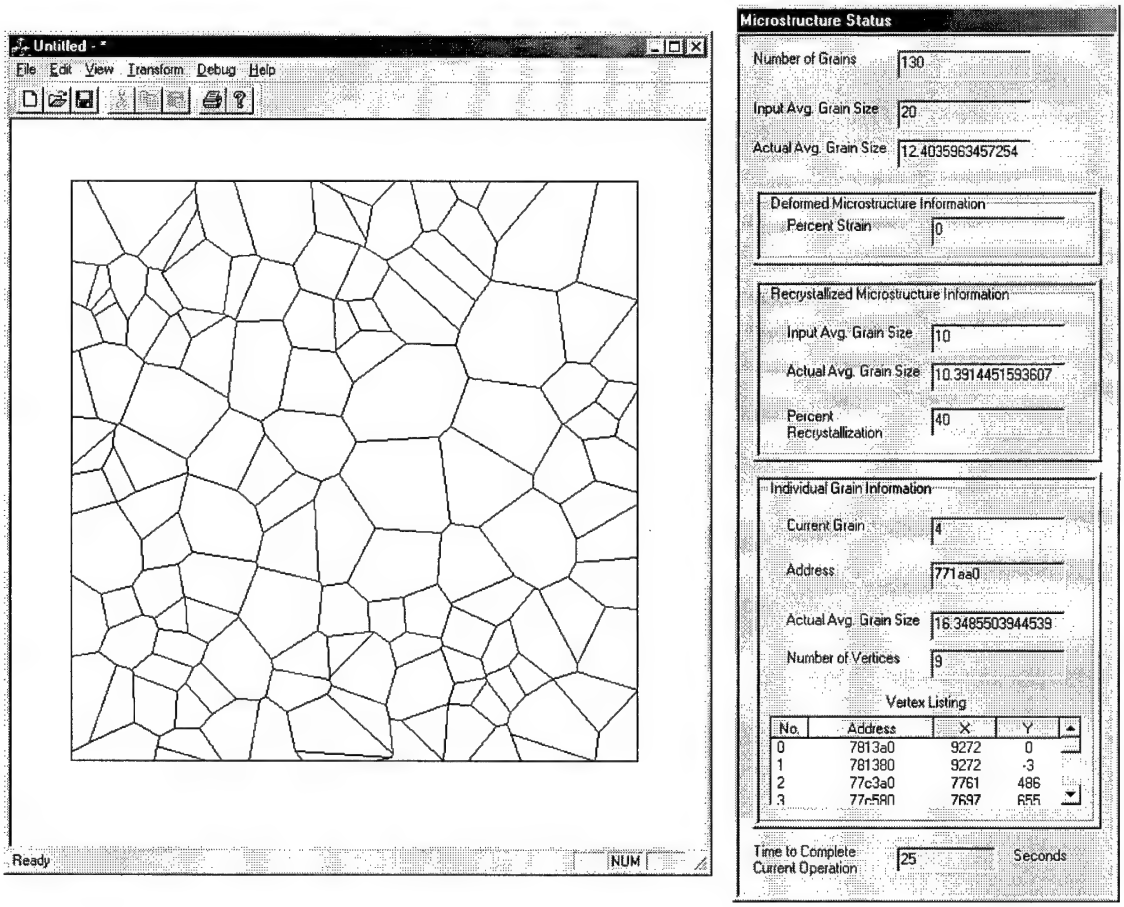


Figure 2-10. Results of Recrystallization Algorithm.

recrystallized grains is then divided by the average recrystallized grain area to calculated the number of recrystallized grains.

The algorithm generates the initial grain centers for the new grains by using a Monte Carlo simulation based on a random number generator algorithm, which allows different probability density functions to be used. As new information about grain

recrystallization mechanics is discovered, new probability density functions may be easily inserted. For this simulation, a uniform distribution function is used for all random number generations. The Monte Carlo generator selects an existing grain for the new grain to be placed within. The Monte Carlo generator then selects a vertex of the existing grain. A region is defined between two points. One point is located at a distance of the square root of the percent recrystallization multiplied by the distance from the centroid to the vertex, and the second point is at the same location as the vertex. A location within this region is found using the Monte Carlo generator. Another location is found along the grain boundary, also using the Monte Carlo generator, between the selected vertex and the next vertex in the selected grains vertex list. A new grain is generated at that location, and the grain is added to a list of new grains separate from the list of original grains. The list of new grains is then added to the master grain list in the microstructure object.

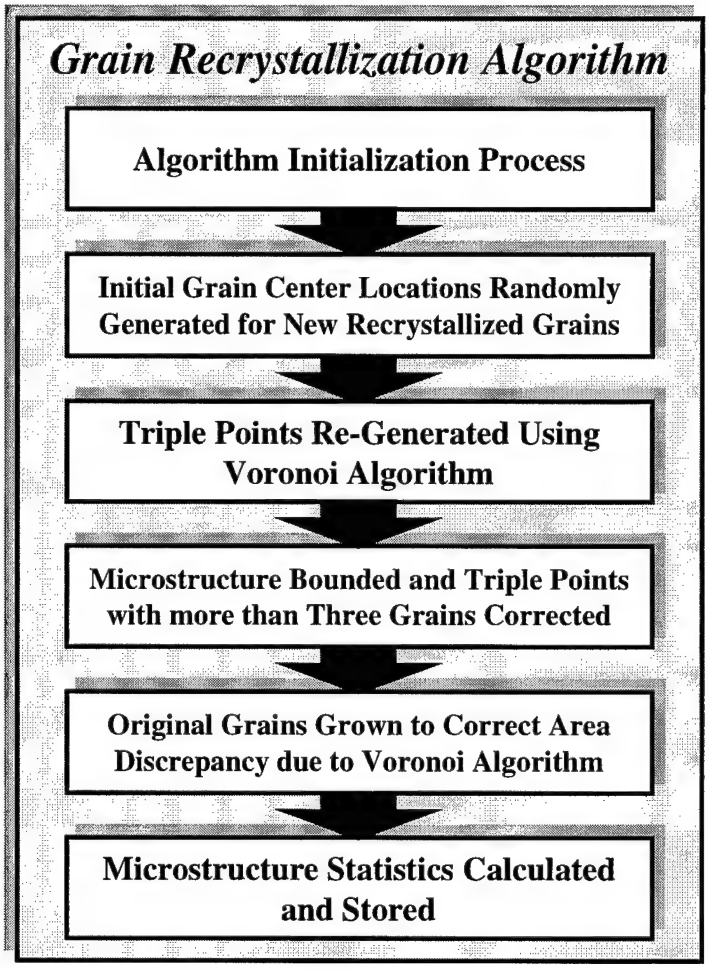


Figure 2-11. Grain Recrystallization Algorithm Process Diagram.

Before the new triple points are generated, the master triple point list is

cleared and all triple points are deleted and removed from memory. The same algorithm used to find the triple points in the initial grain generation algorithm is also used to find the triple points for the grain recrystallization algorithm. The algorithm then sorts the vertices for each grain around the grain's initial center, bounds the grains, calculates the average grain area, and the area and centroids for each grain. The triple points are corrected using the same algorithm that the initial grain generation algorithm uses.

The Voronoi diagram algorithm to find the triple points generates grains with the same average size. However, in the case of a microstructure undergoing recrystallization the average size of the recrystallized grains are not necessarily the same as the average size of the old grains. In order to correct this problem, the algorithm loops through the list of original grains and compares the current size of a grain to its original size. If the new grain size is not within 10 percent of the original size times the percent of recrystallization, a growth factor is calculated by adding 10 percent of the size difference to the current grain size. A pointer to the grain to be grown and the growth factor are passed to the GrowGrain function, explained in more detail in the section on grain growth. After the algorithm has looped through all of the original grains, the algorithm loops through all of the grains and corrects the grains with intersecting boundaries. The process that corrects grain boundary intersection is described in the section on grain growth. The recrystallization algorithm repeats this process until the areas of all of the original grains are within 10 percent of their original size times the percent of recrystallization.

Finally, statistics for the grain recrystallization process are calculated and stored into various data members. The algorithm completion time is calculated and stored in the

microstructure's creation time data member. The actual average grain size is calculated and stored in the actual average grain size data member of the microstructure object. The recrystallized grain size used by the algorithm is stored in the input recrystallized grain size data member, and the actual recrystallized grain size is calculated and stored in the actual recrystallized grain size data member.

2.6 Grain Growth Algorithm

2.6.1 Demonstration Description

In order to start the grain growth algorithm, the user chooses "Grain Growth" from the "Transform" menu. Before the algorithm starts, a window (Figure 2-12) appears which asks for the new grain size. The algorithm uses the new grain size to determine the number of grains to remove in order to grow the microstructure. The microstructure is redrawn in the display window (Figure 2-13) upon completion of the algorithm.

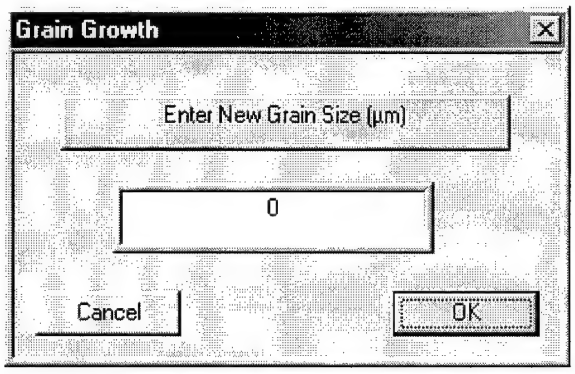


Figure 2-12. Grain Growth Dialog Box.

2.6.2 Algorithm Description

The grain growth algorithm uses a new grain size, supplied by the user, to reduce the number of grains in the microstructure area and therefore, to increase the average grain size. In order to reduce the number of grains to obtain the required average grain size, the algorithm first resizes each grain using an equation based on the number of grain boundaries. After resizing each grain, grain boundary intersection are corrected. Next,

the area of each grain is calculated. If the area of a grain is less than five percent of the average grain area, the grain is removed. After enough grains have been deleted to increase the average grain size to the average grain size specified by the user, the algorithm calculates the algorithm completion time and stores this value in a data member.

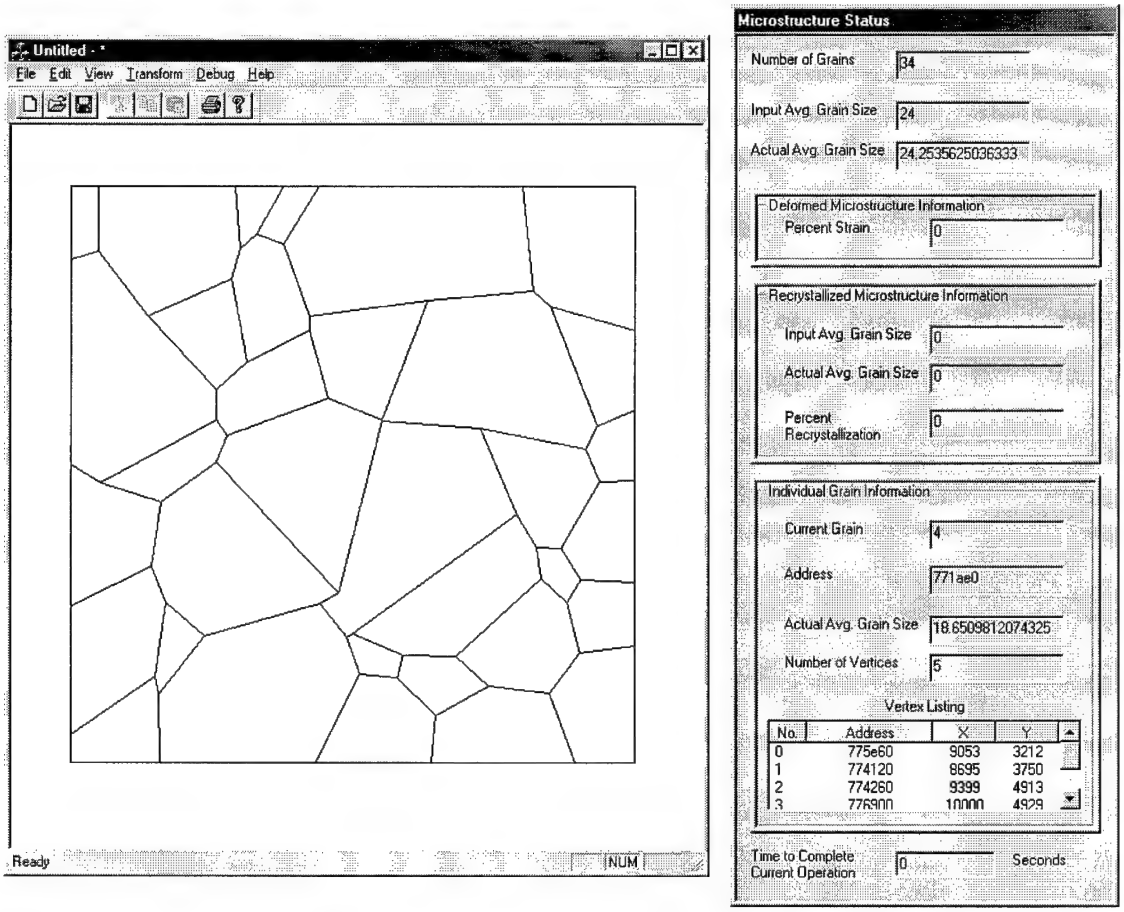


Figure 2-13. Results of the Grain Growth Algorithm.

The first step in the grain growth procedure (Figure 2-14) is to determine the criterion used to stop the grain growth process. This criterion is based on the new average grain size supplied by the user. In grain growth, as the average grain size increase, the number of grains in a given area decreases. The grain growth algorithm determines the new number of grains by dividing the total microstructure area by the new

average grain area. The algorithm performs grains growth steps until the number of grains decreases to the new number of grains.

For each grain growth step, the algorithm searches through the grain list and finds the area of the smallest grain. This value determines the grain growth factor. The list of grains to be removed is also cleared. The algorithm then cycles through the grain list, and for each grain determines the growth factor. The growth factor is calculated by taking the square root of the sum of the current area and the change in area, determined from Equation 2-2, where AreaChange is the change in area, noSides is the number of sides for a specific grain, and minArea is the area of the smallest grain divided by the current area. A pointer to the grain object and the growth factor are passed to the grow grain function.

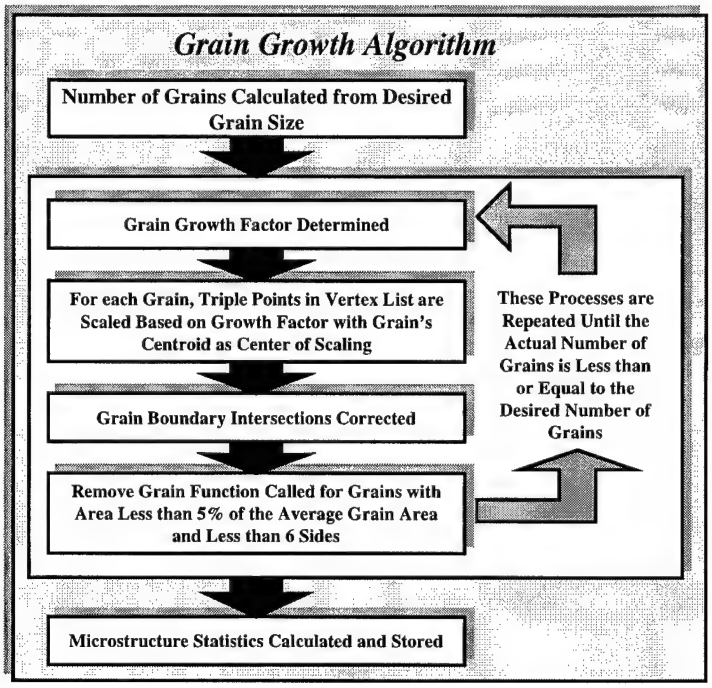


Figure 2-14. Grain Growth Algorithm Process Diagram.

$$AreaChange = 0.5 \cdot \text{minArea}(noSides - 6) \qquad (2\text{-}2)$$

The grow grain function cycles through the grain object's vertex list. For each vertex, the function scales the current location by the growth factor. To scale the vertex location correctly, the function first transposes the reference center from the absolute center at (0,0) to the centroid of the grain. This transposition forces the vertex to be

scaled with respect to the centroid of the grain. After the vertex is scaled, it is transposed again to change the reference center back to the absolute center. After all vertices of the grain have been scaled, the function returns to the grow microstructure algorithm.

After the grains have been grown, the algorithm loops through the grain list and calls the fix grain boundaries function, which fixes boundaries that intersect. The function scrolls through the vertex list of the grain. For each vertex, the function defines the next vertex in the list as the right triple point and defines the previous vertex in the list as the left triple point. The function searches through all of the grains except the current grain, finds the grain that contains the current vertex of interest and the right triple point, and calls that grain the right side grain. Again, the function searches through all of the grains but the current grain, finds the grain that contains the current vertex and the left triple point, and calls that grain the left side grain. Once the left grain or the right grain is found, the triple point directly opposite of the current vertex is found. If the left grain is known, the opposite triple point is the previous vertex from the current vertex in the left grain's vertex list. If the left grain is not known, then the opposite triple point is found to be the next vertex from the current vertex in the right grain's vertex list. After the opposite triple point is found, then the opposite grain is found to be the grain that contains the opposite triple point but is not the left grain or right grain. The left opposite triple point is found to be the triple point that in conjunction with the opposite triple point form the grain boundary between the opposite grain and the left grain. Likewise, the right opposite triple point is the triple point that, along with the opposite triple point, forms the grain boundary between the right and opposite grains. Figure 2-15 shows the defined grains and triple points.

After the six triple points are found, the grain boundaries to be checked are defined. The left grain boundary contains the current vertex and the left triple point. The right grain boundary is defined by the current vertex and the right triple point. The opposite left grain boundary is defined by the opposite triple point and the left grain boundary. The opposite right grain boundary is defined by the opposite triple point and the right grain boundary. Once these grain boundaries have been defined (Figure 2-16), the function tests for intersection between them. If the left grain boundary and the opposite left grain boundary intersect, then a new triple point is created called the new left triple point. If the right grain boundary and the

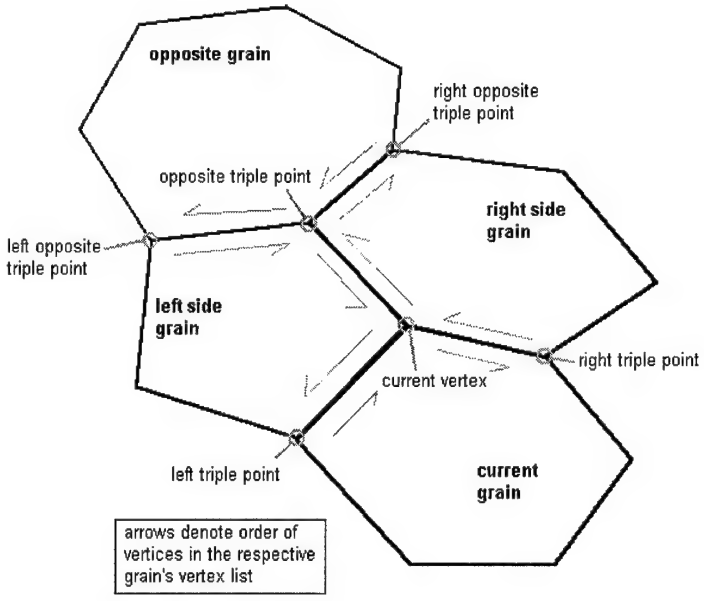


Figure 2-15. Illustration of the Grain and Triple Point Nomenclature for the Process Used to Fix Grain Boundary Intersections.

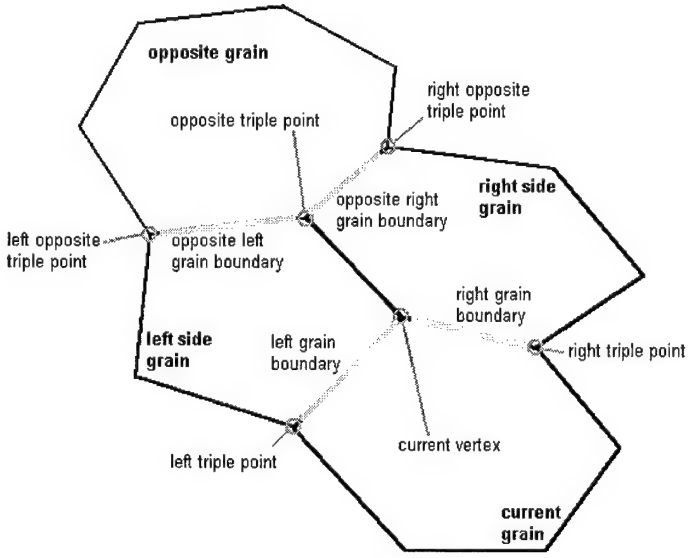


Figure 2-16. Illustration of the Grain Boundary Nomenclature for the Process Used to Fix Grain Boundary Intersections.

opposite right grain boundary intersect, another new triple point called the new right triple point is created. Grain boundary intersection is shown in Figure 2-17.

If the grain boundaries intersect, the new triple points have to be added to the vertex lists of the grains. In the current grain, the current vertex is replaced by the new left triple point, and the new right triple point is added after the current vertex in this grain's vertex list. In the opposite grain, the new right triple point replaces the opposite triple point, and the new left triple point is added after the new right triple point in the opposite grain's vertex list. For the left grain, the opposite triple point is removed from its vertex list, and the new left triple point replaces the current vertex in the left grain's vertex list. In the right grain, the opposite triple point is removed from this grain's vertex list, and the new right triple point replaces the current vertex in the right grain's vertex list. Figure 2-18 shows the results of this process. After the function has cycled through the vertex list, the fix grain boundary function ends, and the program returns to the grow microstructure algorithm.

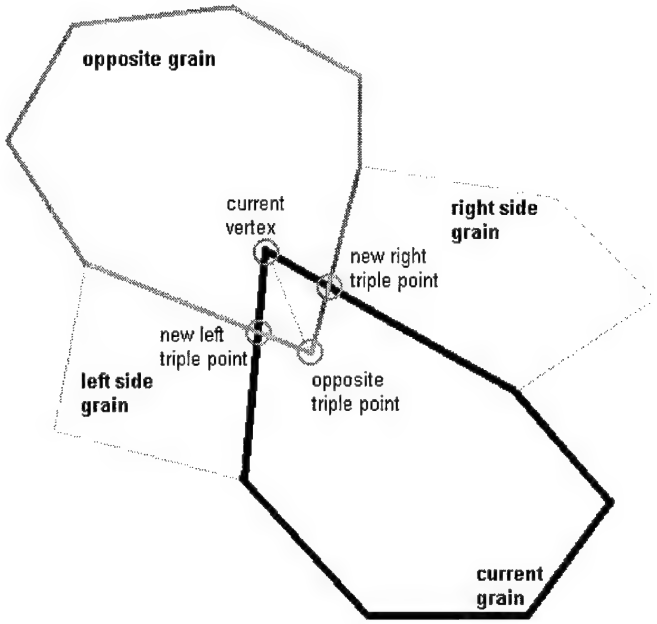


Figure 2-17. Illustration of Grain Boundary Intersection.

The grow microstructure algorithm checks the current area of each grain to determine if the area is less than five percent of the average grain area and the grain has less than six grain boundaries. Each grain that meets these criteria is placed in a list of grains to be removed. The algorithm then cycles through the list of grains to be removed and passes each grain in the list to the remove grain function.

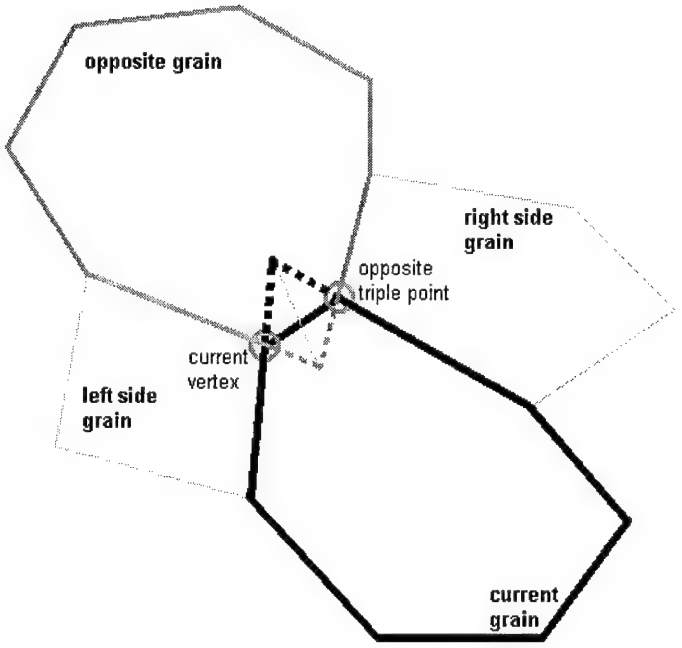


Figure 2-18. Illustration of the Corrections Made Due to Grain Boundary Intersection.

The first operation that the remove grain function performs is to test if the number of vertices of the grain is equal to five. If the grain contains five vertices, the algorithm sweeps through the list of vertices in a clockwise order and finds the vertices that form the shortest grain boundary. The vertices that form this boundary are labeled, in a clockwise order, current triple point and next triple point (Figure 2-19). The function searches through the grain list to find grains, other than the current grain, that satisfies one of three conditions, shown in Figure 2-20. If a grain contains the current triple point and the next triple point (top-left of Figure 2-20), then the current triple point is removed from that grain's vertex list. If a grain contains the next triple point but not the current triple point (top-right of Figure 2-20), then the current triple point is inserted into the

vertex list of the grain before the next triple point. If a grain contains the current triple point but not the next triple point (bottom of Figure 2-20), the next triple point is inserted into the grain's vertex list before the current triple point. The location of the next triple point is moved to the same location as the current triple point. The next triple point is then removed from the current grain's vertex list reducing the number of vertices in the current grain to four shown in Figure 2-21.

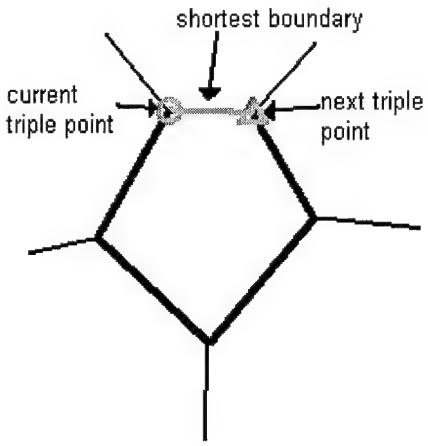


Figure 2-19. Triple Point and Boundary Definitions for the Case of Five-sided Grain Boundary Removal.

The function then checks the grain's vertex list to determine if it has only three vertices. The function then defines the three triple points in the grain's vertex list to be called triple point one, triple point two, and triple point three (Figure 2-22.). The location of triple point one is moved to the location of the centroid of the current grain (Figure 2-23). The algorithm checks all of the triple points to determine if any of

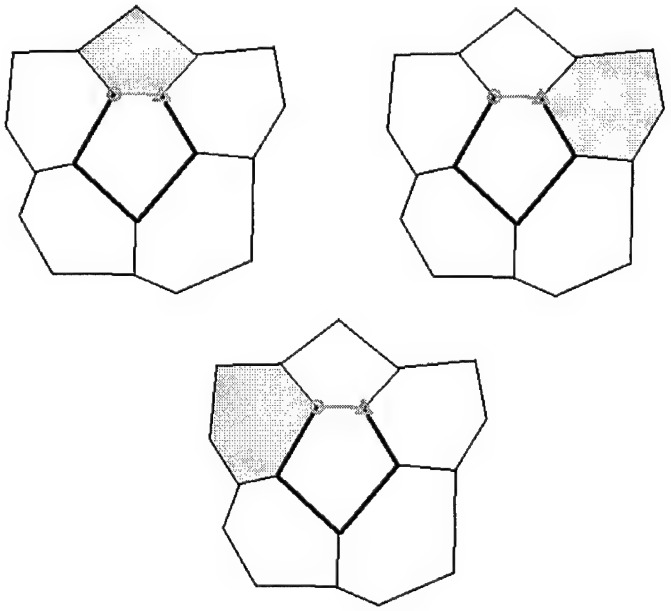


Figure 2-20. Examples of the Three Types of Grains that are Found During the Five-sided Grain Removal Case.

them lie on the boundaries of the microstructure, if one of them does, then triple point one is moves to a location that will maintain the integrity of the overall microstructure boundary (Figure 2-24). The function then searches through the grain list to find a grain, other than the current grain, that meets one of five criteria. If the grain contains triple point one and triple point two (Figure 2-25a), then triple point two is removed from its vertex list. If the grain contains triple point one and triple point three (Figure 2-25b), triple point three is removed from its vertex list. If the grain contains triple point two and triple point three but does not contain triple point one (Figure 2-25c), then triple point two is replaced by triple point one in the grain's vertex list, and triple point three is removed from the vertex list. Triple points two and three are then removed from master triple point list and the current grain's vertex list. The current grain is removed from the microstructure's grain list and deleted from memory. Finally, triple points two and three are deleted from memory and the function returns to the grow microstructure algorithm. Figure 2-26 shows the visual results of these operations.

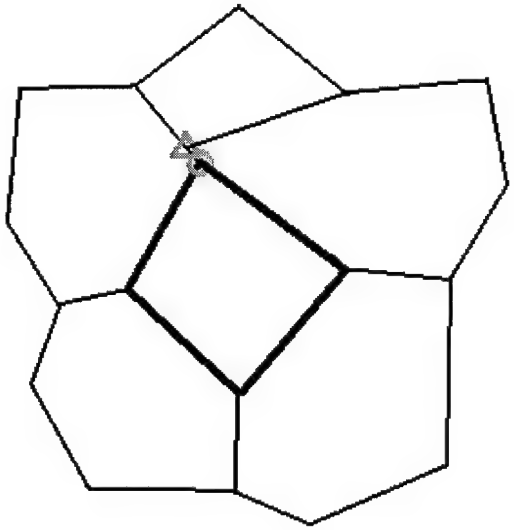


Figure 2-21. Results of Five-Sided Grain Removal.

If the current grain did not have three vertices, then it is checked to determine if it contains four vertices. If the grain has four vertices, then each vertex is labeled, in a clockwise order, triple point one, triple point two, triple point three, and triple point four

respectively. The function then calculates the length of the four grain boundaries and labels them length one, length two, length three, and length four, where length one is opposite of length three and length two is opposite of length four. Figure 2-27 shows the triple points and grain boundaries labeled. The function checks the sum of the length one with length three and the sum of length two with length four.

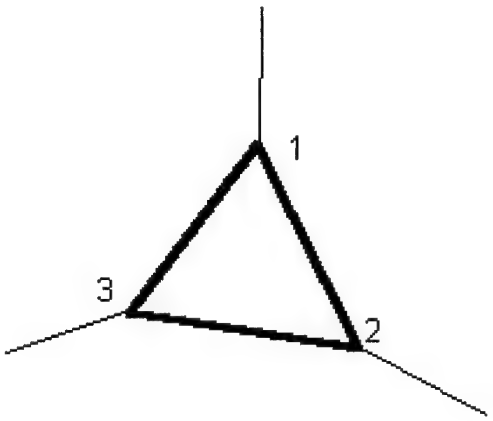


Figure 2-22. Triple Points Defined for the Case of Three-sided Grain Removal.

The function finds the mid points of each of the grain boundaries with the smaller of the two sums and moves one of the two triple points that compose one of the smaller grain boundaries to the mid point for each grain boundary (Figure 2-28). The function then removes the other point that defines one of the two smaller grain boundaries from all of the grains and deletes it from the master triple point list and memory. The triple points that were moved are added to the grains with the longer grain boundaries, if they do not already contain those triple points. Finally, the current grain is removed from the master grain list, deleted from memory, and the function returns to the grow microstructure algorithm. Figure 2-29 shows an example of the results of this procedure.

If the current grain only contains two vertices, then the function cycles through the vertex list of the current grain. For each vertex in the list, the function finds the vertex in another grain's vertex list and removes it from that list. The vertex is then removed from the current grain's vertex list and the master triple point list and deleted

from memory. After all vertices have been removed from the current grain, the current grain is removed from the microstructure's grain list and deleted from memory.

After the algorithm has finished removing grains, it cycles through the grain growth step until the number of grains is less than or equal to the desired number of grains as determined by the new grain size. Once this criterion is reached, the grain growth cycle ends. The actual average grain size for the microstructure is calculated and stored in a data member within the microstructure object. Finally, the operation time is computed and stored in a data member completing the execution of this algorithm.

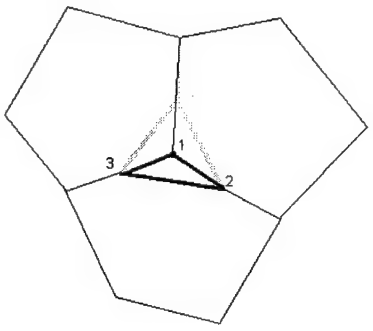


Figure 2-23. Illustration of Relocation of Triple Point in Three-sided Grain Removal.

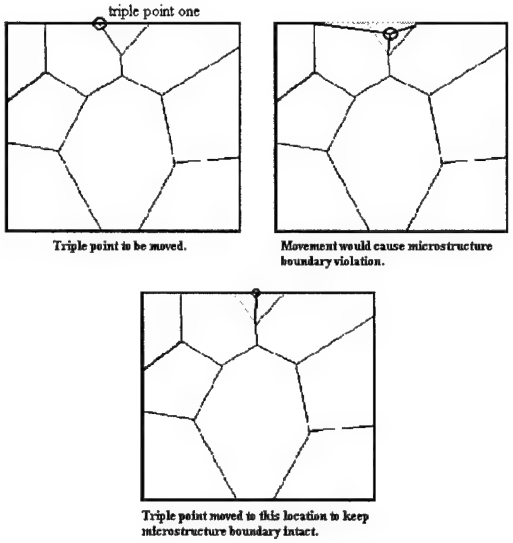


Figure 2-24. Example of Correcting a Triple Point that Lies on the Microstructure Boundary for the Three-sided Grain Removal Process.

The grain growth algorithm uses these procedures to manage growth of the virtual microstructure within the visual computer simulation system. The growth algorithm together with the other algorithms described in this chapter are evaluated in Chapter 3 to determine the time to produce a solution, the accuracy or error of the results, and the validation of the solution as compared with other published data.

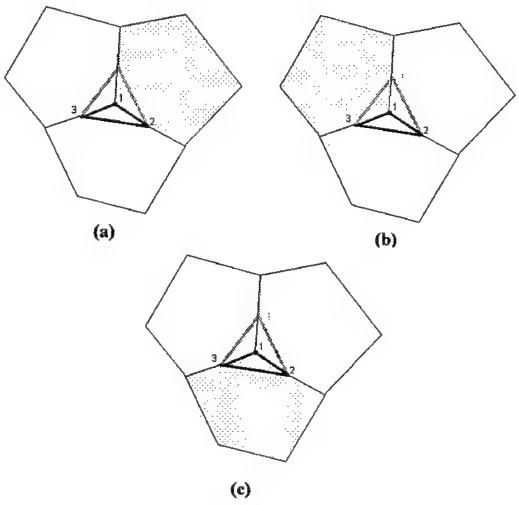


Figure 2-25. Triple Point Cases for Three-sided Grain Removal.

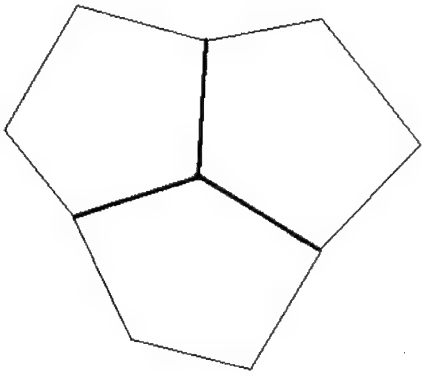


Figure 2-26. Results of Three-sided Grain Removal.

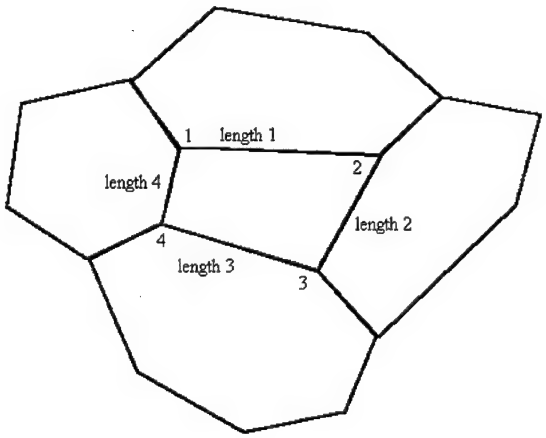


Figure 2-27. Illustration of the Triple Points and Grain Boundaries Involved in Four-sided Grain Removal.

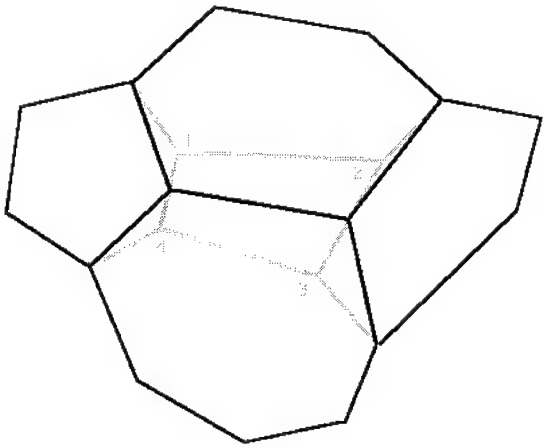


Figure 2-28. Illustration of Triple Point Adjustments Performed in Four-sided Grain Removal.

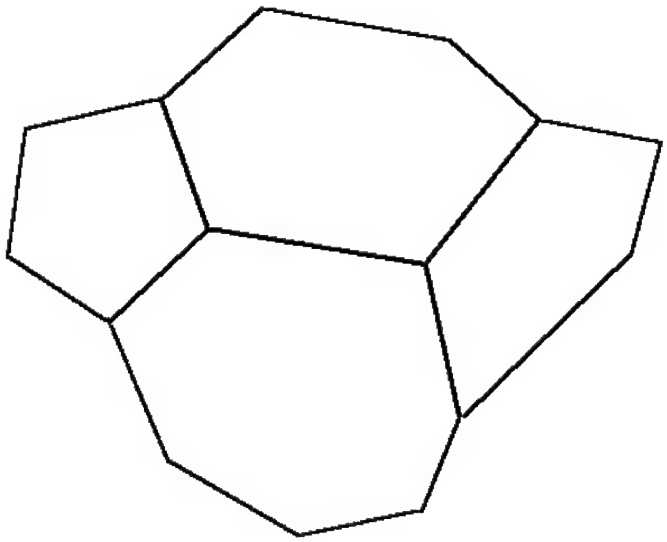


Figure 2-29. Four-sided Grain Removal Results.

CHAPTER 3

SIMULATION TESTING

3.1 Introduction

This chapter presents the results of tests performed on the microstructure simulation. These tests were performed on an IBM-compatible PC with a 200 MHz MMX Intel Processor and 98 MB of RAM. The microstructures are presented for both qualitative and quantitative comparison. These tests show performance of the algorithms as measured by operation time, comparisons between the criteria set by the user and the actual results, and comparison with microstructure data from the literature. Each algorithm is presented in a separate section. The first algorithm discussed is the initial microstructure algorithm, followed by the grain deformation algorithm, then the grain recrystallization algorithm, and finally the grain growth algorithm.

3.2 Initial Microstructure Algorithm Test Results

This section describes the results of the initial microstructure tests. The first of these tests show the time taken for the algorithm to finish depending on the number of grains to be simulated. The next test compares the grain size distribution for various grain sizes and then compares the grain size distribution for various numbers of grains. The last test compares the grain size distributions of ten simulated microstructures to the grain size distribution of a microstructure found in the literature. Finally, a

microstructure is generated with the simulation and visually compared with microstructures generated by other methods found in the literature.

3.2.1 Algorithm Time to Solution

The purpose of the time-to-solution test is to show the factors in the initial microstructure algorithm that affect the amount time the algorithm takes to complete. The two factors controlled by the user are the number of grains and the initial grain size. According to the tests performed, the number of grains is the only user-controlled factor that affects the time to solution. The initial grain size does not affect the time to solution. Two series of tests were preformed in order to show this.

The first series of tests were performed to show the dependency of the time to solution on the number of grains. For this series, microstructures with different number of grains but the same grain size (100 microns) were generated. The number of grains started at 50 grains and ranged to 150 grains by steps of ten and also included tests for 200, 250, 300, 400 and 500 grains. Five microstructures were generated for each different number of grains, and the time for computer-generated results of the five microstructures were averaged. The average time for each of the number of grains are presented in Figure 3-1.

The second series of tests showed that the time to solution was not dependent on the average grain size. For this series, additional microstructures, in sets of five, were generated for grain sizes of 10, 20, 50, and 100 microns were generated for grain numbers of 50, 100, 200, and 400. The averages of each of these sets were compared and presented in Figure 3-2. This table shows that time to solution appears to not depend on the grain size.

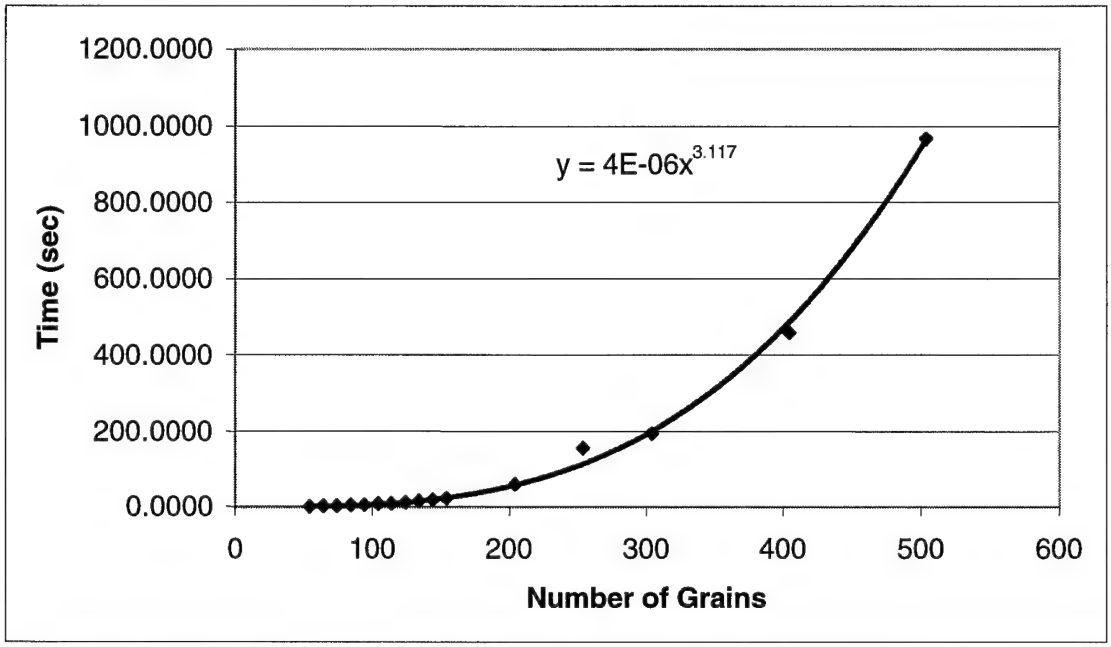


Figure 3-1. Time to Solution as a Function of the Number of Grains.

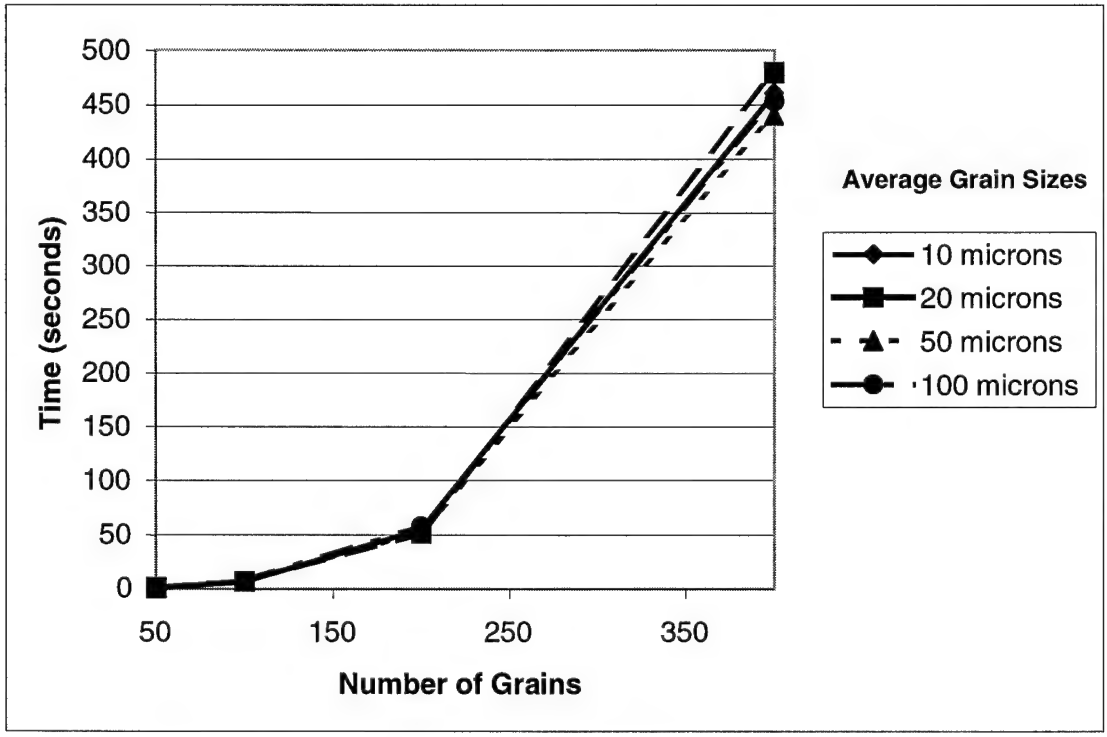


Figure 3-2. Time to Solution Comparison for Different Average Grain Sizes.

3.2.2 Grain Size Distribution Comparisons

The purpose of the grain size distribution comparison is to show how the average grain size and the number of grains affect the grain size distribution. The first series of tests compare distributions of grain size for simulated microstructures with the same average grain size, but different numbers of grains. The second series compare grain size distributions of simulated microstructures with the same number of grains, but different average grain sizes.

For this series of tests, microstructures are generated with the same average grain size of 100 microns. The grain size distributions of microstructures with grain numbers of 50 grains, 100 grains, 200 grains, 300 grains, and 500 grains are compared. Figure 3-3

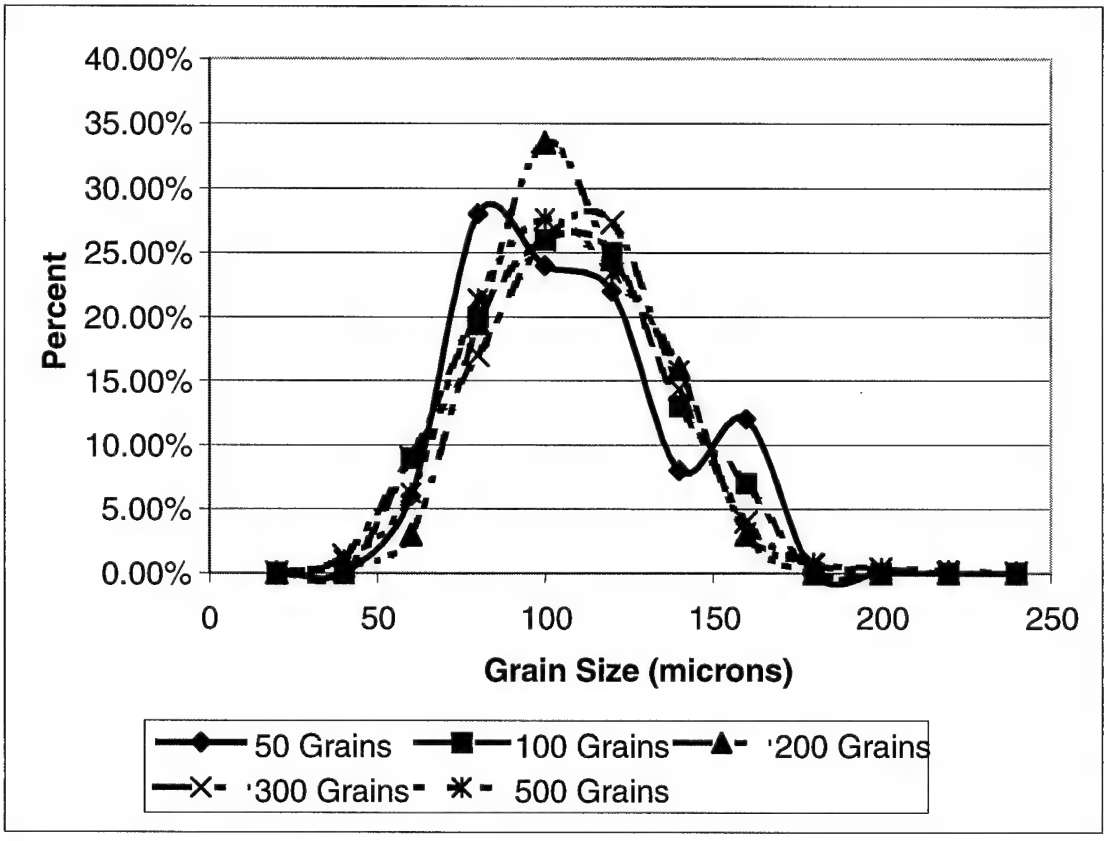


Figure 3-3. Grain Size Distributions for Different Numbers of Grains with an Average Grain Size of 100 Microns.

shows the grain distributions resulting from this series of tests. The results show that the grain size distribution appears to be relatively independent of the number grains.

The next series of tests involve generating microstructures with the same number of grains set at 100 grains, but the average grain size is changed. Microstructures are generated with average grain sizes of 10, 20, 30, 50, and 100 microns to show the effect of average grain size on the grain size distribution. The results of this series of tests are shown in Figure 3-4.

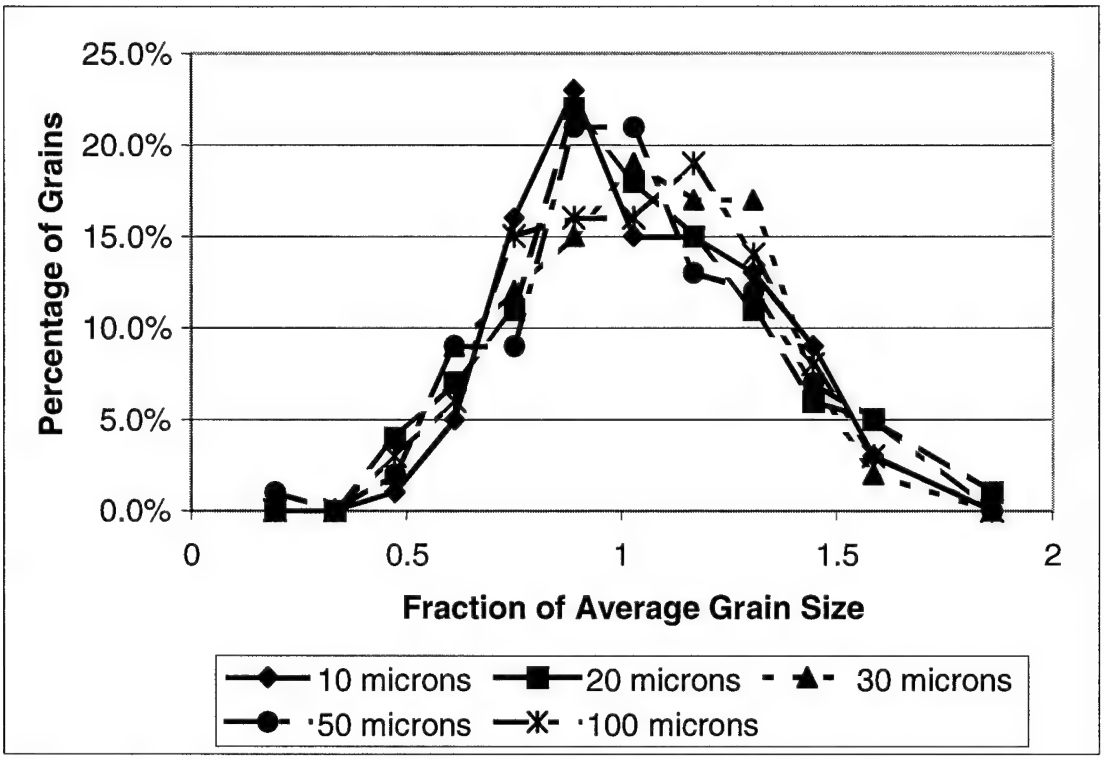


Figure 3-4. Grain Size Distributions for Different Average Grain Sizes.

3.2.3 Comparison with Reference Grain Size Distribution

This subsection shows the results of grain size distributions of simulated microstructures as compared with grain size distributions from published sources. The first set of tests compares the grain size distribution of a simulated microstructure with

the grain size distribution of a microstructure measured using ASTM standards. The last set compares the grain area distribution of microstructures created by this computer program with grain area distributions of microstructures simulated by other computer programs.

This first test compares the grain size distribution of a simulated microstructure with the grain size distribution of a microstructure analyzed by the Saltykov area method described by Vander Voort (1984b). For this test, the simulation is used to generate five separate microstructures with the ASTM grain size number of 8.5 given by the Vander Voort (1984b). The ASTM grain size number is converted into a grain size with units in microns. Since 470 grains were observed in the reference, the simulation generates 470 grains. The resulting grain size distributions from the five simulated microstructures and the reference are compared in Figure 3-5 (Vander Voort, 1984b).

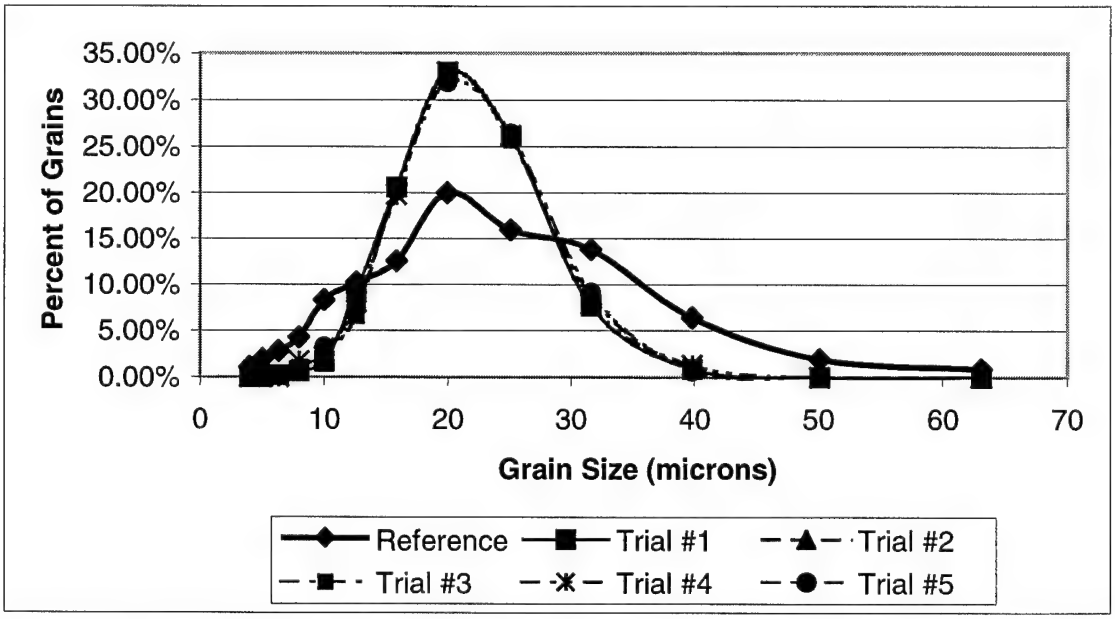


Figure 3-5. Grain Size Distribution Comparisons Between Simulated Microstructures and Size Distribution Analysis from Reference (Vander Voort, 1984).

The second test compares the grain area distribution from a simulated microstructure with the distribution based on a simulation performed by Frost and Thompson (1986). Frost and Thompson generated a microstructure with 46,000 grains using a site saturation Voronoi polygon distribution method. The grain areas were normalized by dividing the individual grain areas by the average grain area. The grain area distribution found by Frost and Thompson is shown in Figure 3-6. Because of limits to the computation speed of the algorithm in this thesis, a microstructure with 500 grains

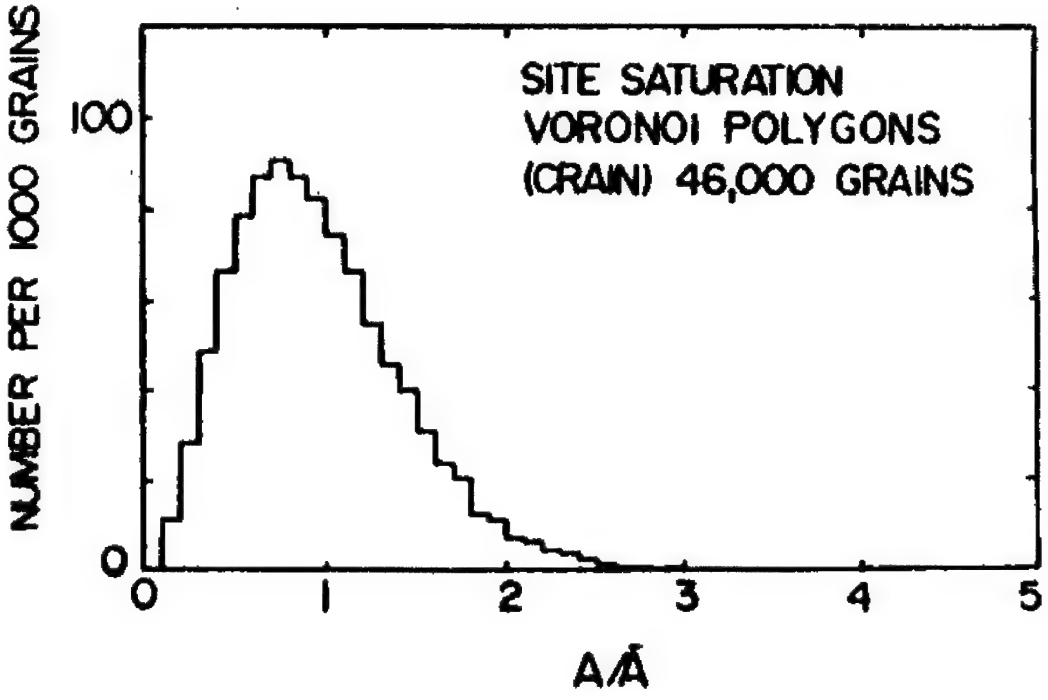


Figure 3-6. Grain Area Distribution of a 46000 Grain Microstructure Created by the Site Saturation Voronoi Polygon Method (Frost and Thompson, 1986).

and a grain size of 100 microns was generated. The resulting grain sizes were converted into grain areas by multiplying the square of the grain diameters by ¼ π. The areas were then divided by the average grain area and presented in Figure 3-7 using the same scale as Figure 3-6 (Frost and Thompson, 1986). There is an apparent similarity in the shape between the two curves.

3.2.4 Visual Comparison with Reference Microstructures

This section visually compares a microstructure generated by the simulation with microstructures generated by a simulation developed by Frost and Thompson (1986).

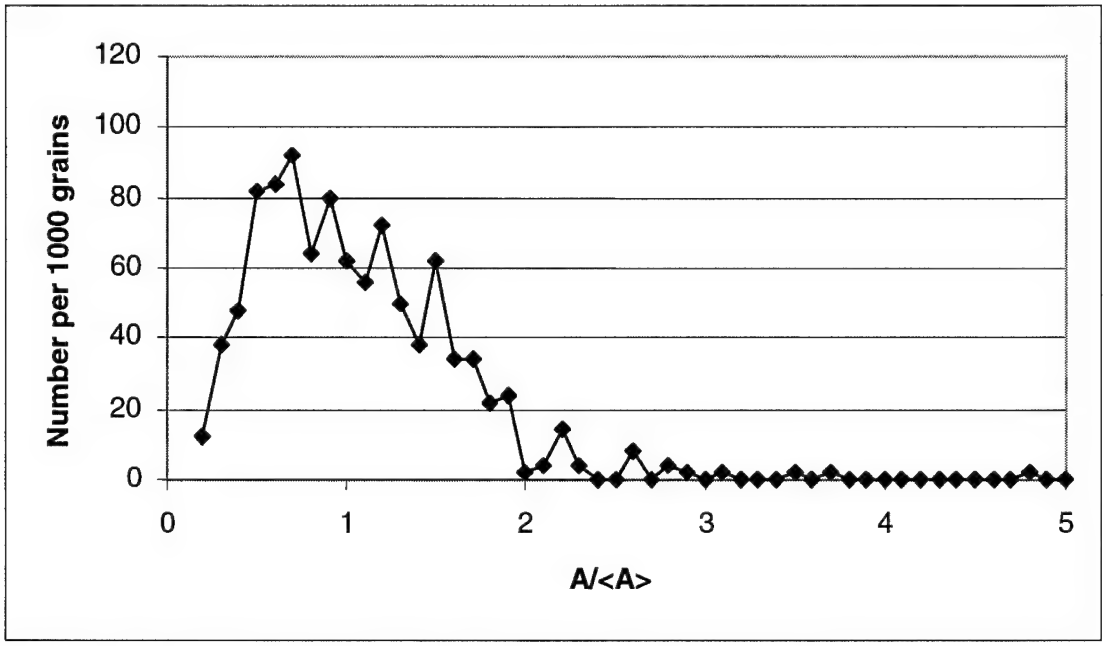


Figure 3-7. Normalized Grain Area Distribution of 500 Grains.

The microstructure generated by the designed simulation has 470 grains at an average grain size of 19 microns. The simulated microstructure is compared with a microstructure generated by the site saturation Voronoi method and by the Johnson Mehl method. Figure 3-8 shows the comparison.

3.3 Grain Deformation Algorithm Test Results

This section describes the results of the grain deformation algorithm tests. The first test shows the algorithm's time to solution as it depends on the number of grains and percent strain. The next test shows the error associated with the average deformation of all the individual grains as compared with the expected input grain deformation. For this

test, differing numbers of grains and percent strains are examined. The final subsection visually compares actual metals that have been deformed with the simulation to show grain elongation that occurs in grain deformation.

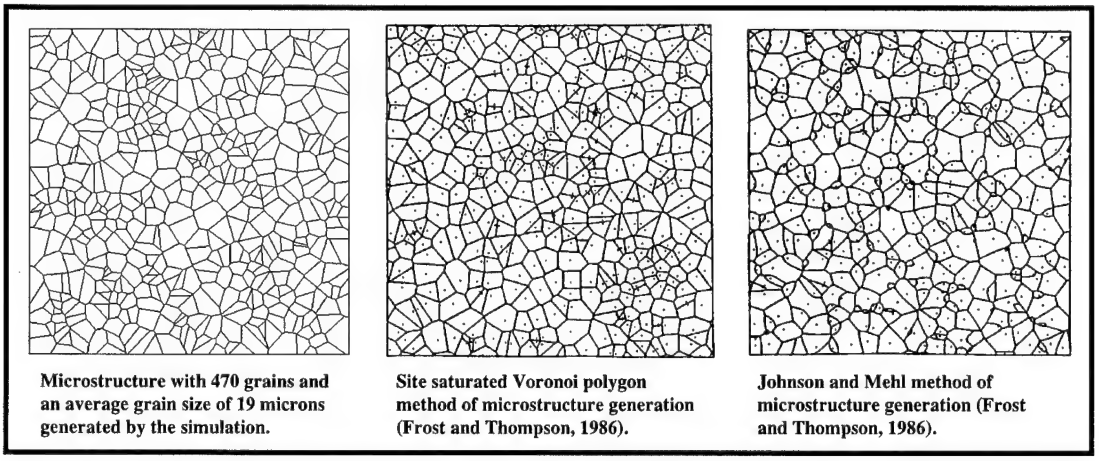


Figure 3-8. Visual Comparison Between Simulated Microstructures.

3.3.1 Algorithm Time to Solution

This section contains the results of the tests performed to determine how much time the algorithm took to arrive at a final microstructure that satisfied the inputs. For this test, four separate initial microstructures were used. Each initial microstructure was generated with the same average grain size of 100 microns but different numbers of grains (50, 100, 150, and 200 grains). Deformations ranging from 10 to 90 percent strain with a 10 percent step size were performed ten times for each microstructure, and the results were compiled and are shown in Figure 3-9.

3.3.2 Error Between Expected Results and Actual Results

This subsection describes the tests that determine the reliability of the microstructures resulting from the deformation algorithm. Two sets of tests are performed in this subsection. This set of tests compares the error between the expected

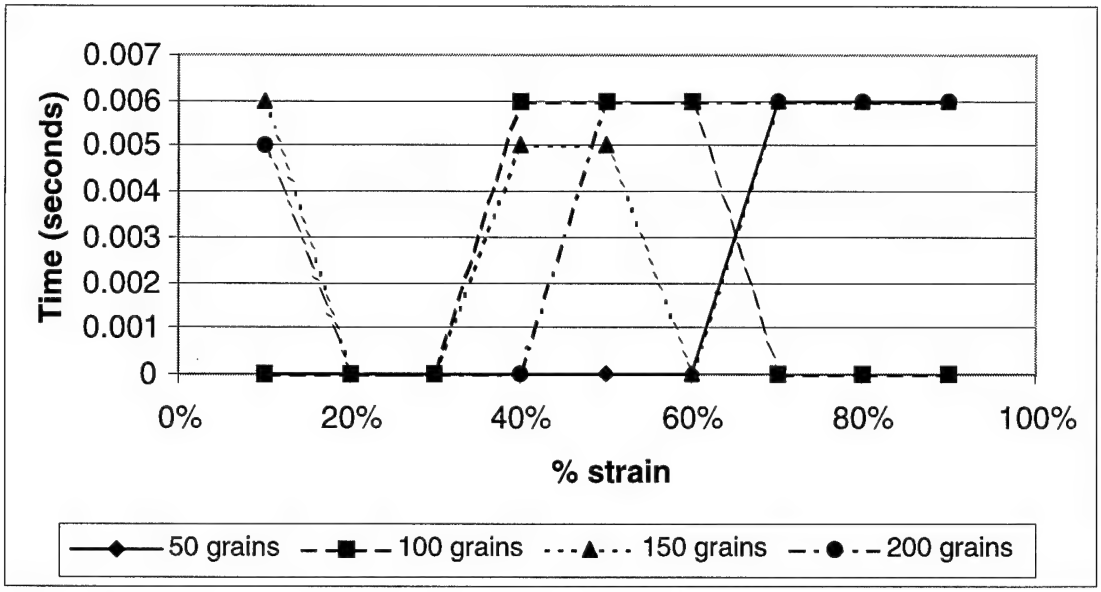


Figure 3-9. Time to Solution for Grain Deformation Algorithm at Different Numbers of Grains.

input percent deformation, and the actual deformation. An initial microstructure of 100 microns is generated for four sets of grain numbers (50, 100, 150, and 200 grains). The microstructures are deformed with a percent deformation ranging from 10 to 90 percent at a step size of 10 percent. Each microstructure test is performed ten times. The actual deformation is determined by dividing the new aspect ratio into the old aspect ration for each individual grain. The resulting individual deformations are averaged, and the average is compared with the expected input deformation and shown in Figure 3-10.

3.3.3 Visual Deformation Comparison with Reference Microstructures

A qualitative comparison between actual deformed microstructures and simulated deformed microstructures is performed in this section. For this comparison, a microstructure is simulated with 200 grains and an initial grain size of 100 microns. The microstructure is deformed at different percent strains ranging from 10 to 60 percent at increments of 10 percent strain. Figure 3-11 shows the microstructures generated from

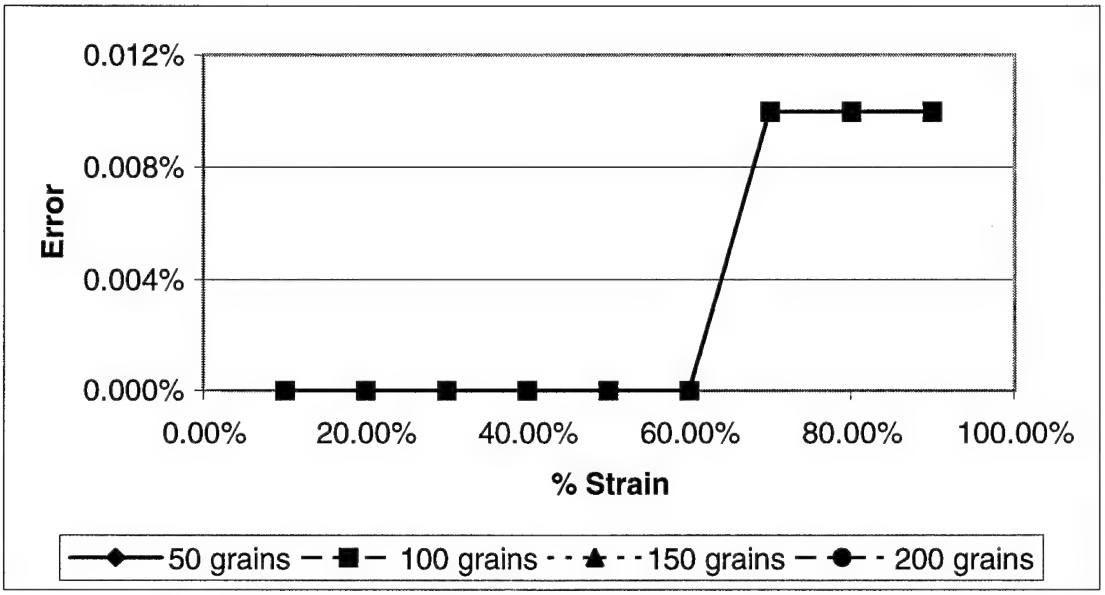


Figure 3-10. Difference Between Actual and Expected Percent Deformation.

the simulation. The simulated microstructure is compared to a sample of capped 1008 steel (Figure 3-12) which has also been deformed at different percent strains ranging from 10 to 60 percent in increments of 10 percent strain taken from the Metals Handbook, 9th Edition (American Society for Metals, 1985).

3.4 Grain Recrystallization Algorithm Test Results

This section describes the results of the grain recrystallization algorithm tests. The first subsection shows the algorithm's time to solution with a variable percent recrystallization and variable recrystallized grain size. The error between the expected recrystallized grain size and actual recrystallized grain size and between the expected percent recrystallization and actual recrystallization is presented in another subsection. The last subsection visually compares actual recrystallized metal in a reference with simulated recrystallized microstructures.

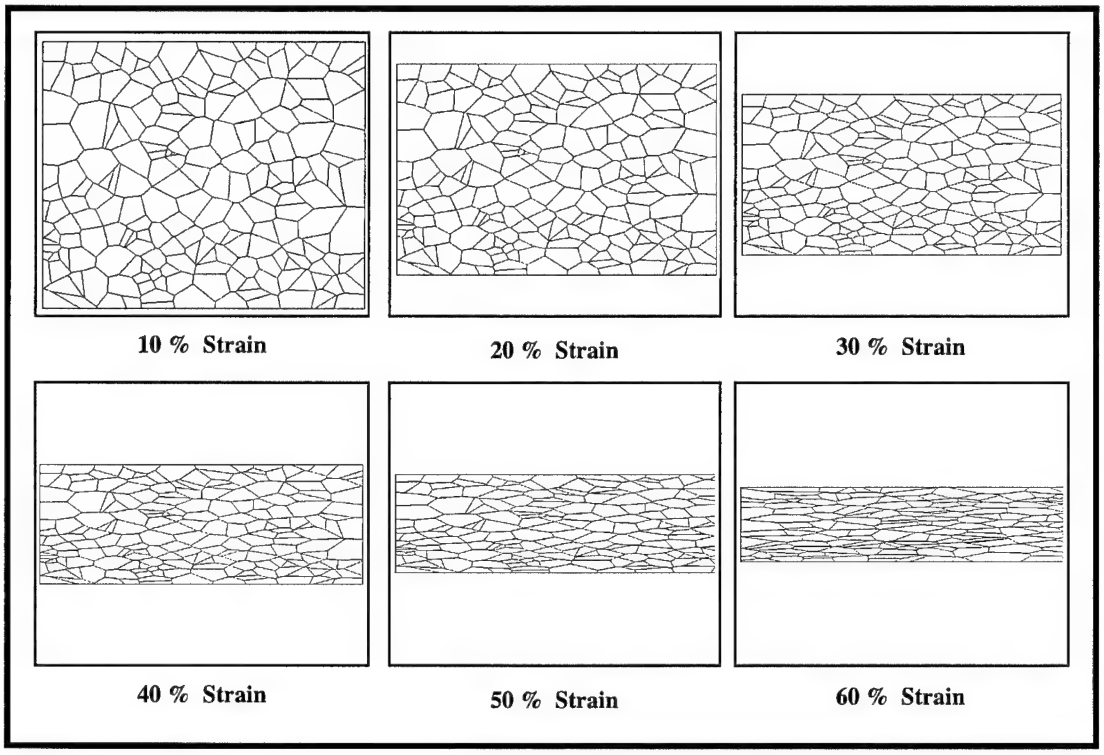


Figure 3-11. A Simulated Microstructure with 100 Grains Shown and an Average Grain Size of 100 Microns. Shown at Different Percent Reductions.

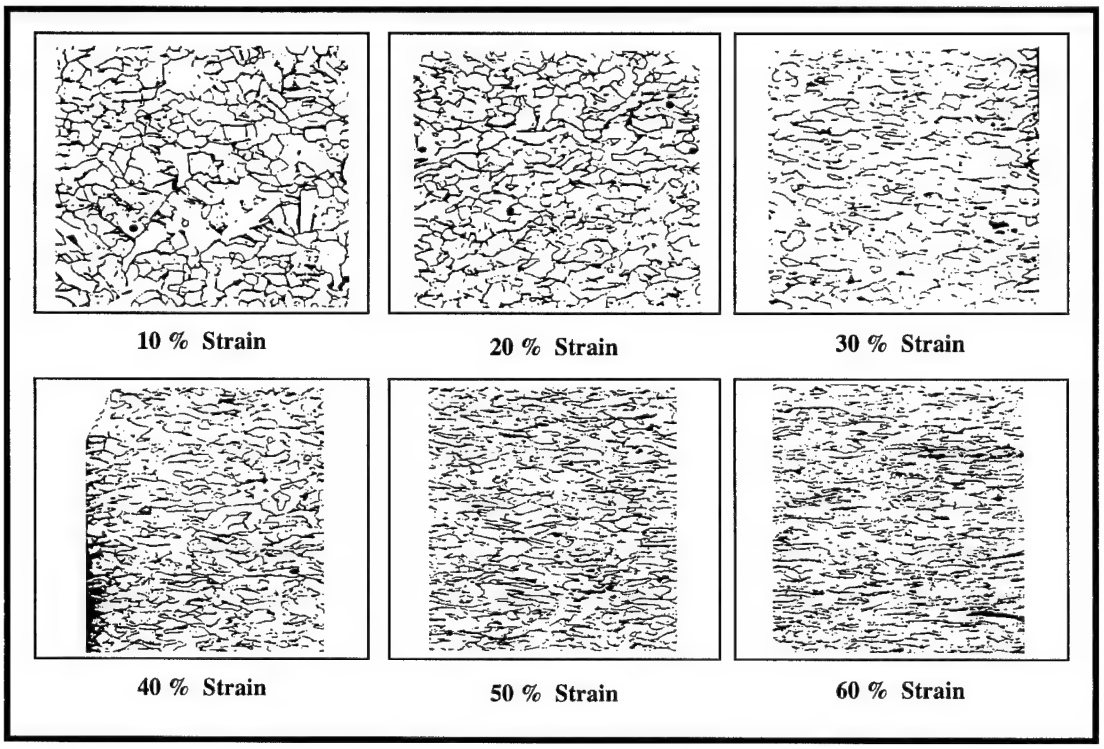


Figure 3-12. Capped 1008 Steel, Finished Hot, Coiled Cold, then Hot Rolled from a Thickness of 3 mm. Shown at Different Percent Reductions (American Society for Metals, 1985).

3.4.1 Algorithm Time to solution

The tests in this subsection show the recrystallization algorithm's time to solution as it is influenced by process parameters. The first series of tests show the time to solution with a variable percent recrystallization and a constant recrystallized grain size. The recrystallized grain size is varied in the next series of tests while the percent recrystallization is kept constant. These tests show how the percent recrystallization and recrystallized grain size effect the time to achieve a graphical solution.

For this first series of tests, a microstructure with an initial grain size of 100 microns and displaying 50 grains is used in order to maintain a consistent starting point for the recrystallization tests. The initial microstructure is recrystallized with a constant recrystallized grain size of 50 microns. The simulation is run with a percent recrystallization of 10 to 80 percent in 10 percent increments. For each percent recrystallization, ten trials are performed to get ten times that are averaged together. The results of these tests are shown in Figure 3-13.

The initial microstructure used in the previous series with an initial grain size of 100 microns and 50 grains is used for another series of tests. The percent recrystallization is set at a constant 30 percent for all of these tests while recrystallized grain sizes ranging from 30 to 100 microns in increments of 10 microns are selected. Ten trials are executed for each grain size and the results are averaged. Figure 3-14 shows the results of this series of tests.

3.4.2 Comparison between Expected and Actual Recrystallized Grain Size and Percent Recrystallization

The tests described in this subsection show how well the simulation displays a recrystallized microstructure with the characteristics specified by the user. The first set

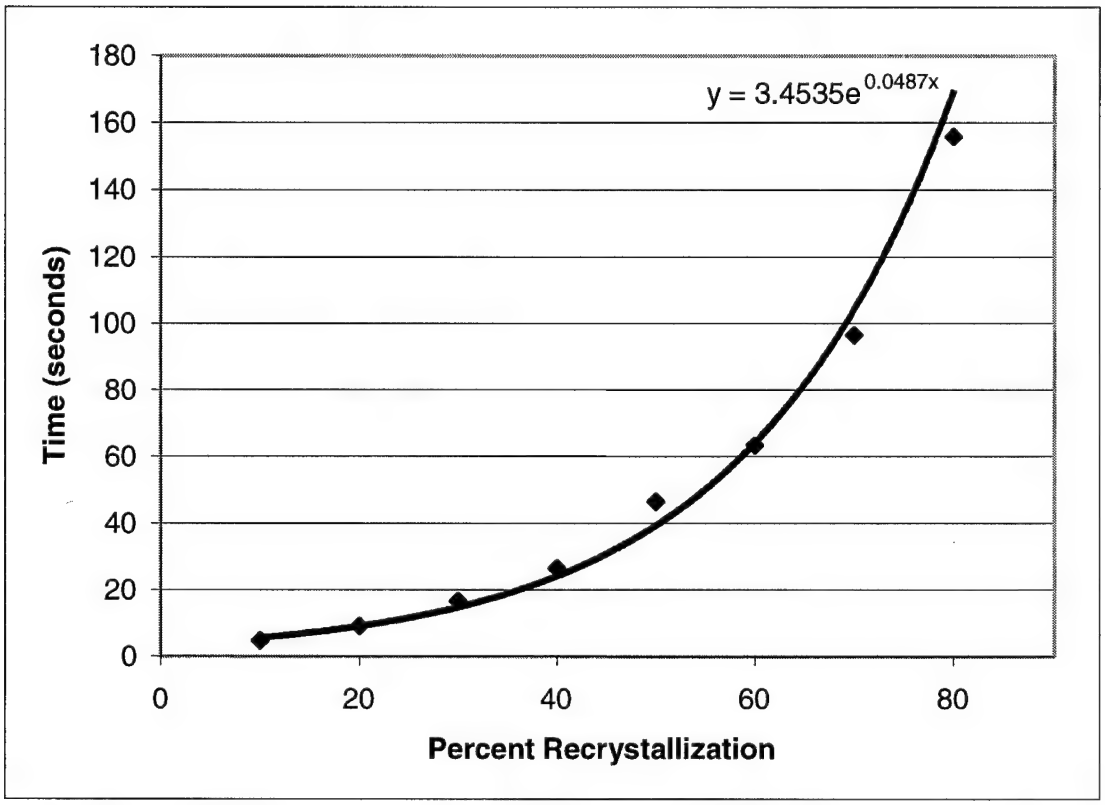


Figure 3-13. Time to Solution for Recrystallization at Recrystallized Grain Size of 50 Microns.

of tests shows how the difference between the expected and average recrystallized grain size and percent recrystallization is affected by different percent recrystallizations with a constant recrystallized grain size. The second set of tests uses a constant percent recrystallization to show how different recrystallized grain sizes effect the difference between the expected and average recrystallized grain size and percent recrystallization.

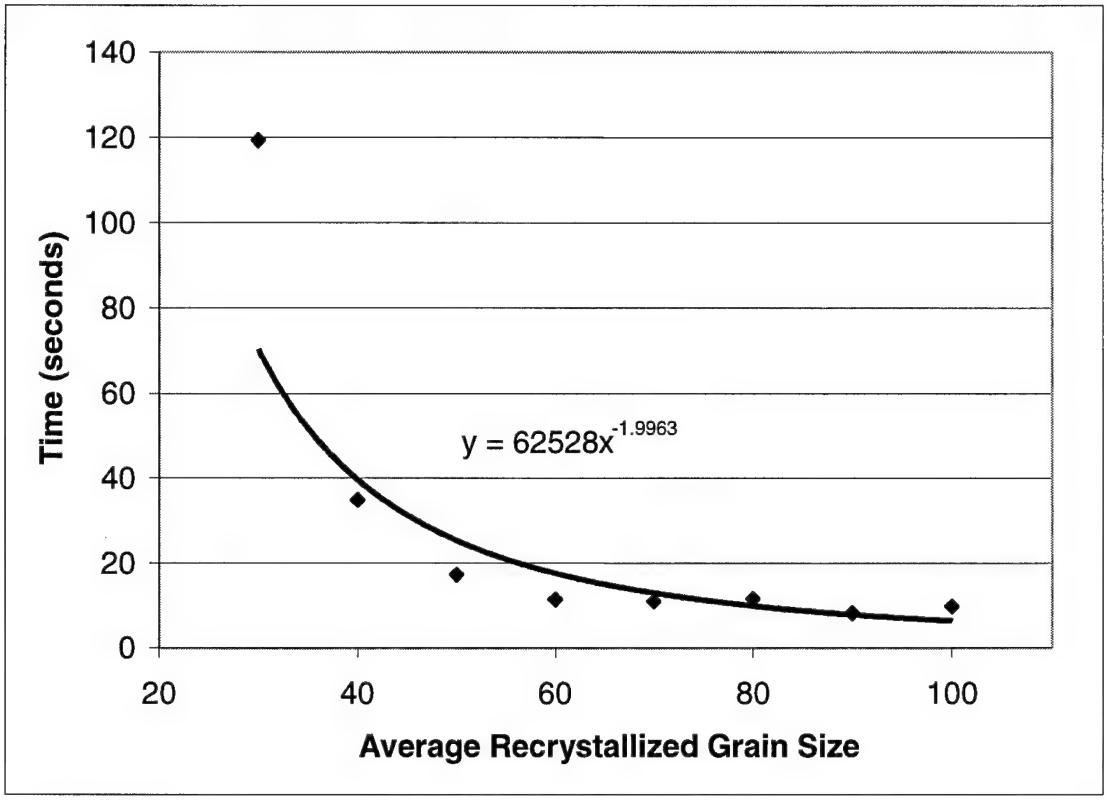


Figure 3-14. Time to Solution for Different Average Recrystallized Grain Sizes at the Same Percent Recrystallization of a Microstructure with 100 Micron Initial Average Grain Size.

For the first set of tests, a microstructure with an initial grain size of 100 microns displaying 50 grains is used for consistent starting point for the tests. From this starting point, the initial microstructure is recrystallized with a constant recrystallized grain size of 50 microns. Ten simulations are performed at different percent recrystallizations ranging from 10 to 80 percent in 10 percent increments. The actual recrystallized grain sizes and percent recrystallizations are compared with the expected recrystallized grain sizes and percent recrystallizations. Figure 3-15 shows the error between actual and expected recrystallized grain size, and Figure 3-16 shows the percent difference between actual and expected percent recrystallization.

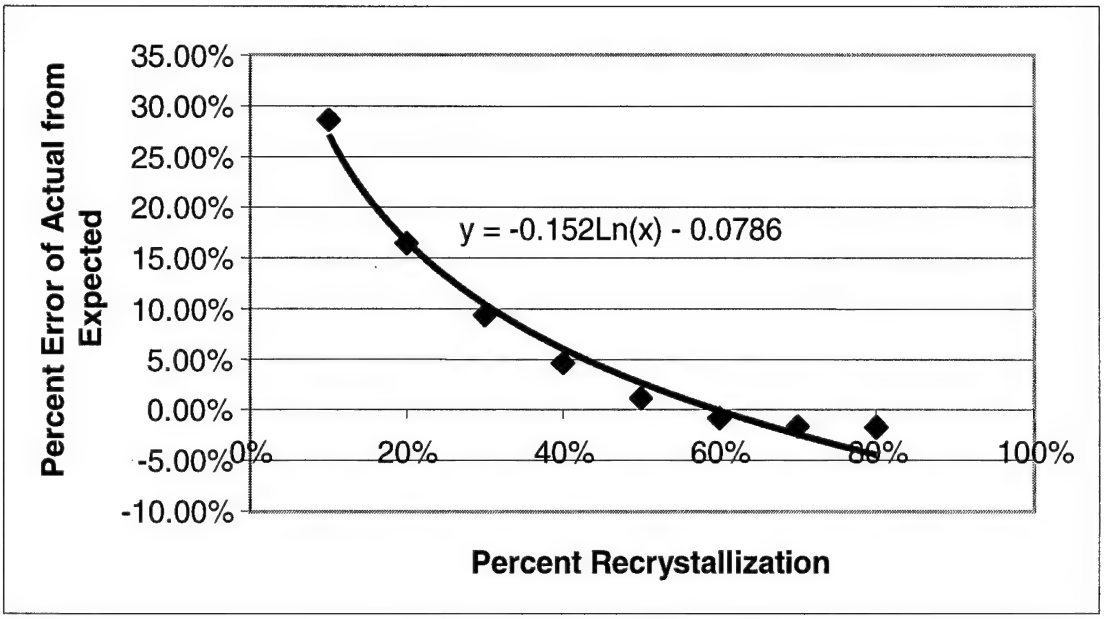


Figure 3-15. Error Between Actual and Expected Grain Size for Different Percent Recrystallizations at Constant Grain Size of 50 Microns.

Like the first set of tests, a microstructure with an initial grain size of 100 microns and displaying 50 grains is used for the second set of tests. The initial microstructure is recrystallized at a constant percent recrystallization of 30 percent. Again, ten simulations are run for recrystallized grain sizes from 30 to 100 microns in increments of 10 microns. Figure 3-17 shows the error between the actual and expected recrystallized grain sizes, and Figure 3-18 shows the percent difference between the actual and expected percent recrystallizations.

3.4.3 Effect of Recrystallized Grain Size and Percent Recrystallization on Grain Size Distribution

The two sets of tests in this subsection explore how the grain size distribution is affected by different recrystallized grain sizes and percent recrystallizations. The first set of tests shows the effect of the recrystallized grain size on grain size distribution. The

effect of percent recrystallization on grain size distribution is shown in the second set of tests.

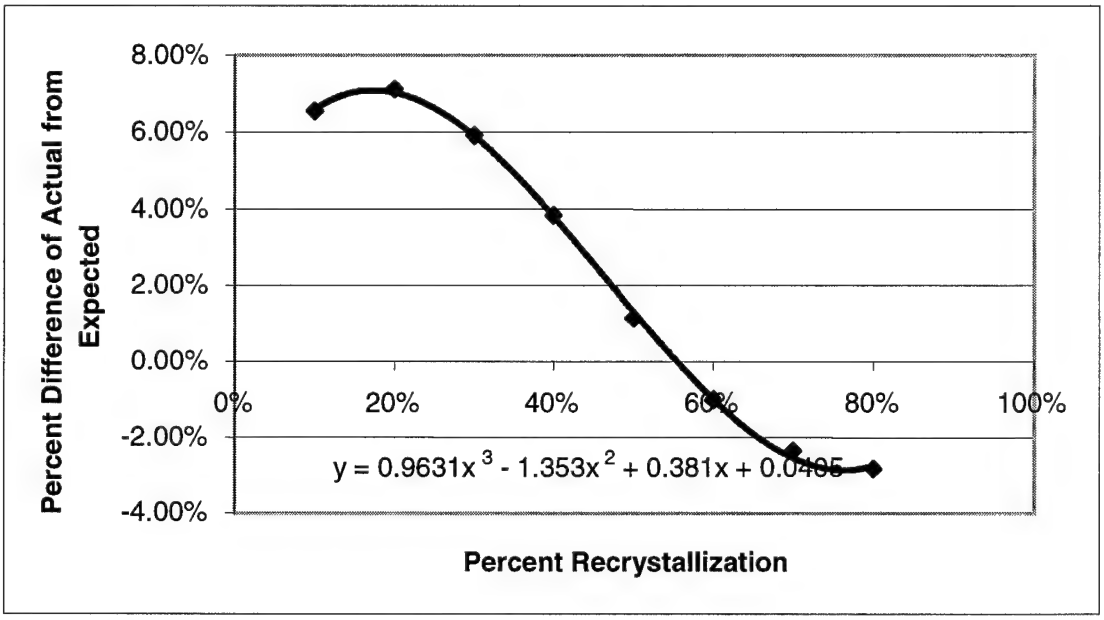


Figure 3-16. Percent Difference Between Actual and Expected Percent Recrystallization for Different Grain Sizes at a Constant Recrystallized Grain Size of 50 Microns.

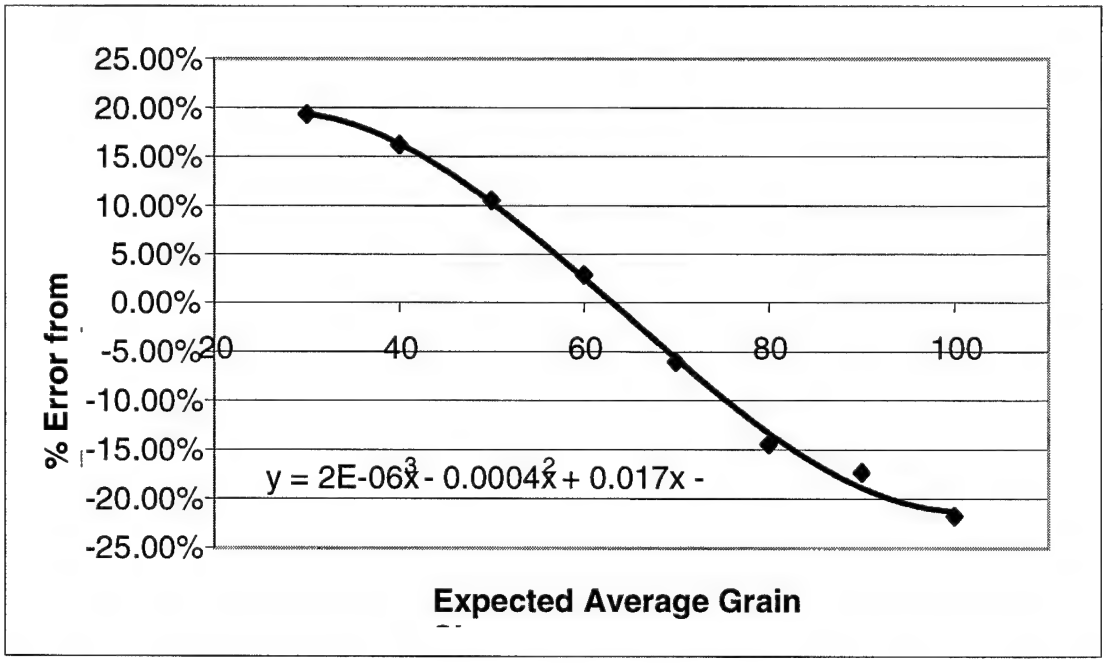


Figure 3-17. Error Between Expected and Actual Recrystallized Average Grain Size for Different Recrystallized Grain Sizes at Constant 30 Percent Recrystallization.

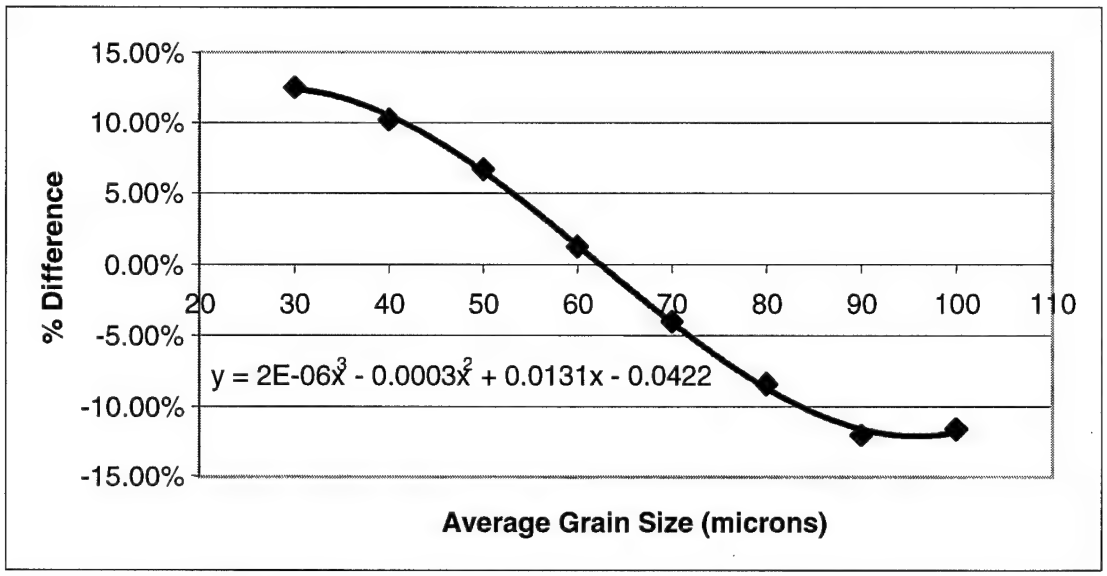


Figure 3-18. Percent Difference Between Actual and Expected Percent Recrystallization for Different Grain Sizes at a Constant 30 Percent Recrystallization.

In the first series of tests, a microstructure with an initial grain size of 100 microns, displaying 50 grains, is generated as the starting point. From this starting point, the initial microstructure is recrystallized with a constant percent recrystallization of 30 percent. A simulation is generated for each recrystallized grain size of 30, 40, 60, 80, and 100 microns. Figure 3-19 compares the grain size distributions for the five different recrystallized grain sizes to show how the recrystallized grain size affects the grain size distribution.

The second series of tests show how the percent recrystallization affects the grain size distribution. An initial microstructure is generated with an average grain size of 100 microns and 50 grains displayed. This initial microstructure is recrystallized with a constant recrystallized grain size of 50 microns, while percent recrystallizations of 10, 20, 40, 60, and 80 percent are examined. The grain size distributions for the five different percent recrystallizations are shown in Figure 3-20.

3.4.4 Visual Comparison between Recrystallized Grain Size Distribution Simulated and from Reference

This subsection contains the results of comparisons between the grain size distributions formed by the simulation and by microstructures from reference. Simulations were performed for microstructures displaying 20 grains. One simulation

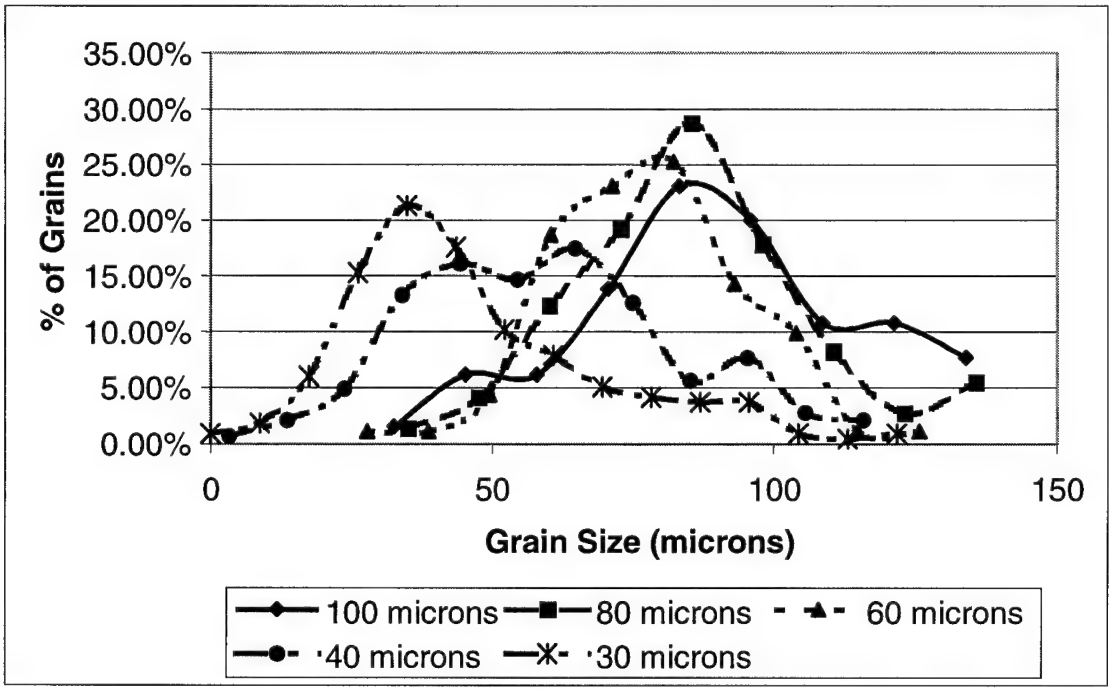


Figure 3-19. Grain Size Distribution for Different Recrystallized Grain Sizes at 30 Percent Recrystallization.

uses an initial grain size of 100 microns and recrystallizes the microstructure with a recrystallized grain size of 20 microns and a percent recrystallization of 60 percent. The second simulation uses an initial grain size of 70 microns and recrystallizes the microstructure with a recrystallized grain size of 20 microns and a percent recrystallization of 90 percent. The last simulation uses an initial grain size of 50 microns and recrystallizes the microstructure with a recrystallized grain size of 10 microns and a percent recrystallization of 40 percent. The simulated recrystallized

microstructures are compared with recrystallized microstructures from published sources in Figure 3-21.

3.5 Grain Growth Algorithm Test Results

This section describes the results of the grain growth algorithm tests. The first subsection shows algorithm time to solution with a variable number of initial grains and

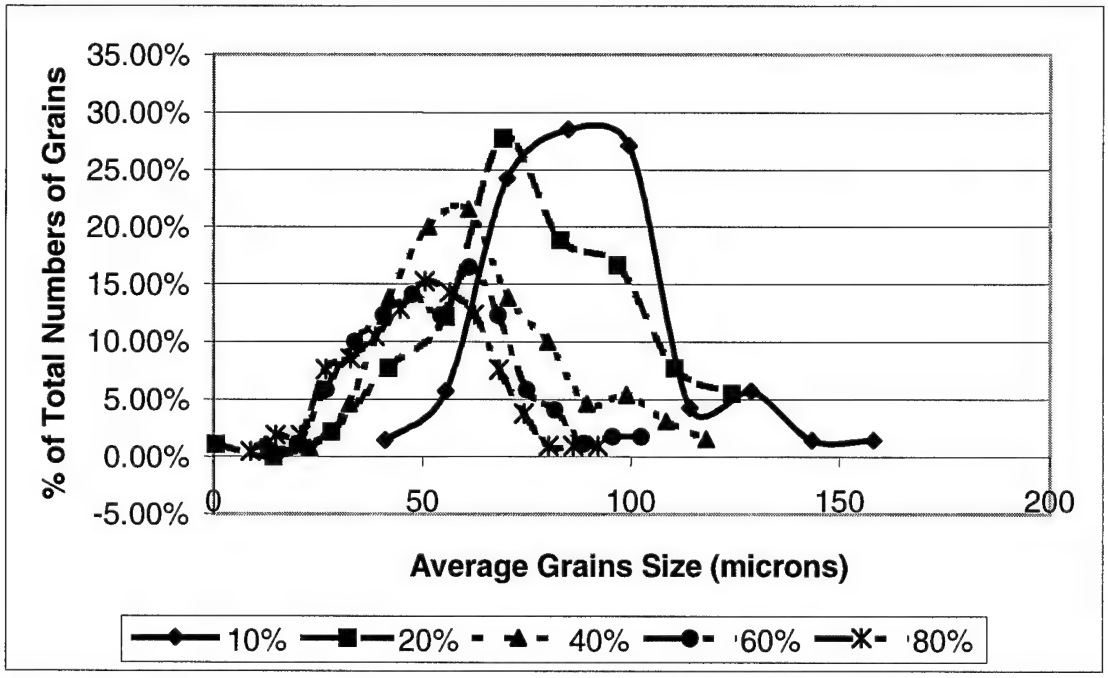
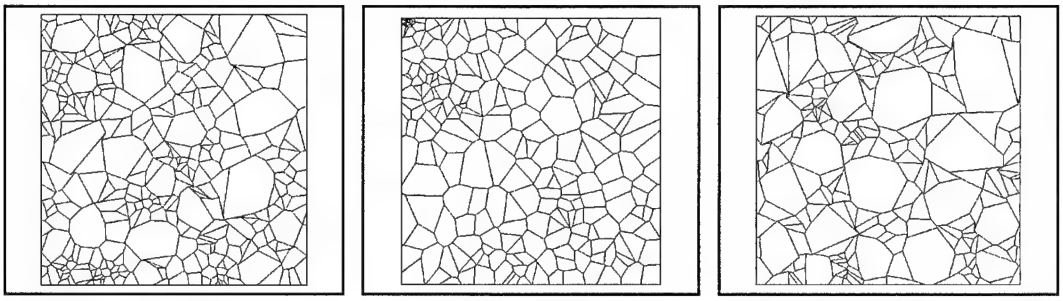


Figure 3-20. Grain Size Distributions for Different Percent Recrystallizations with a Recrystallized Grain Size of 50 Microns.

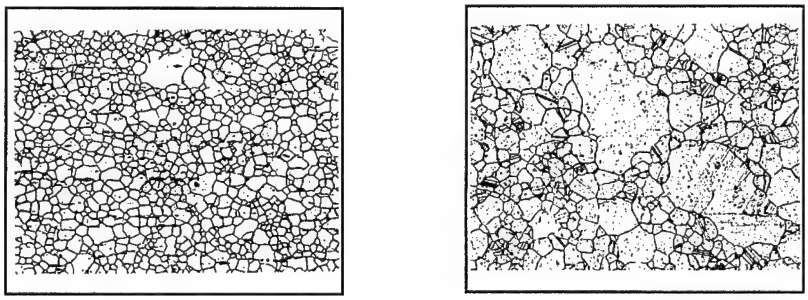
constant new grain size. Also shown in this subsection is the time to solution with a constant number of initial grains and a variable new grain size. The comparison between the expected new average grain sizes and the actual average grain sizes shows how well the simulation matches the users required inputs. A series of grain size distributions are presented showing the moving of the distribution curve as the grain size changes. Graphs of the grain size distributions normalized with respect to the average grain size are compared for a microstructure, as it is grown. These graphs are compared with



Simulated with 20 initial grains at 100 microns. Recrystallized at 60% recrystallization with recrystallized grain size of 20 microns.

Simulated with 20 initial grains at 70 microns. Recrystallized at 90% recrystallization with recrystallized grain size of 20 microns.

Simulated with 20 initial grains at100 microns. Recrystallized at 40% recrystallization with recrystallized grain size of 10 microns.



Example of a well-dispersed duplex grain structure in a low-carbon steel at 97.4% recrystallization and a recrystallized grain size of 12 microns (Vander Voort, 1984b).

A duplex grain structure of a Pyromet 31 nickel-base superalloy with 85.4% recrystallization and a recrystallized grain size of ~88 microns (Vander Voort, 1984a).

Figure 3-21. Comparison Between Recrystallized Microstructures From Simulation and From Reference.

information from the published sources. Finally, grain growth simulated with this program is compared with grain growth simulated by a program from the literature.

3.5.1 Algorithm Time to solution

The tests performed in this section are performed to show the time to solution for the grain growth algorithm. The time to solution is measured for varying new average grain sizes. For each test, an initial microstructure of 100 grains with an average grain size of 100 microns is used. Tests are performed for new average grain sizes ranging from 105 microns to 150 microns at a step size of 5 microns. Ten simulations are

performed for each new average grain size and averaged. The average times are plotted versus the new average grain size and shown in Figure 3-22.

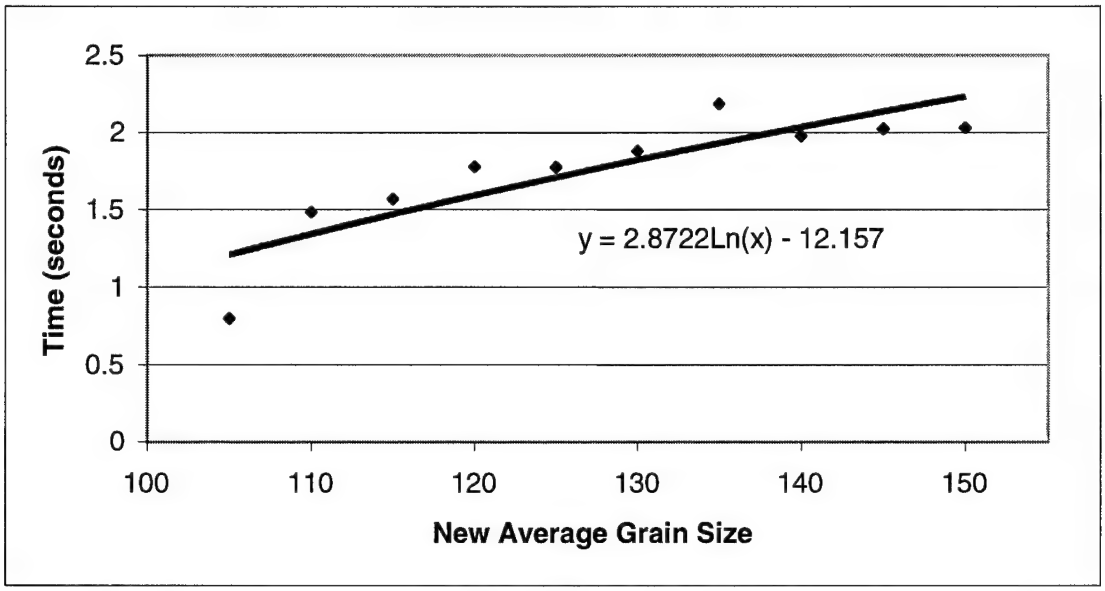


Figure 3-22. Time to Solution for Grain Growth as a Function of the New Average Grain Size.

3.5.2 Expected Average Grain Size Compared to Actual Average Grain Size

The comparison between the expected new average grain size and actual new average grain size shows the algorithm's reliability. For this test, an initial microstructure with and average grain size of 100 microns and 100 grains is used. Ten simulations are performed at each new average grain size ranging from 105 microns to 150 microns in steps of 5 microns. The actual new average grain size for the ten simulation are averaged and compared with the user-specified new average grain size. The resulting percent differences between the actual and expected average grain size for each different expected new average grain size are shown in Figure 3-23.

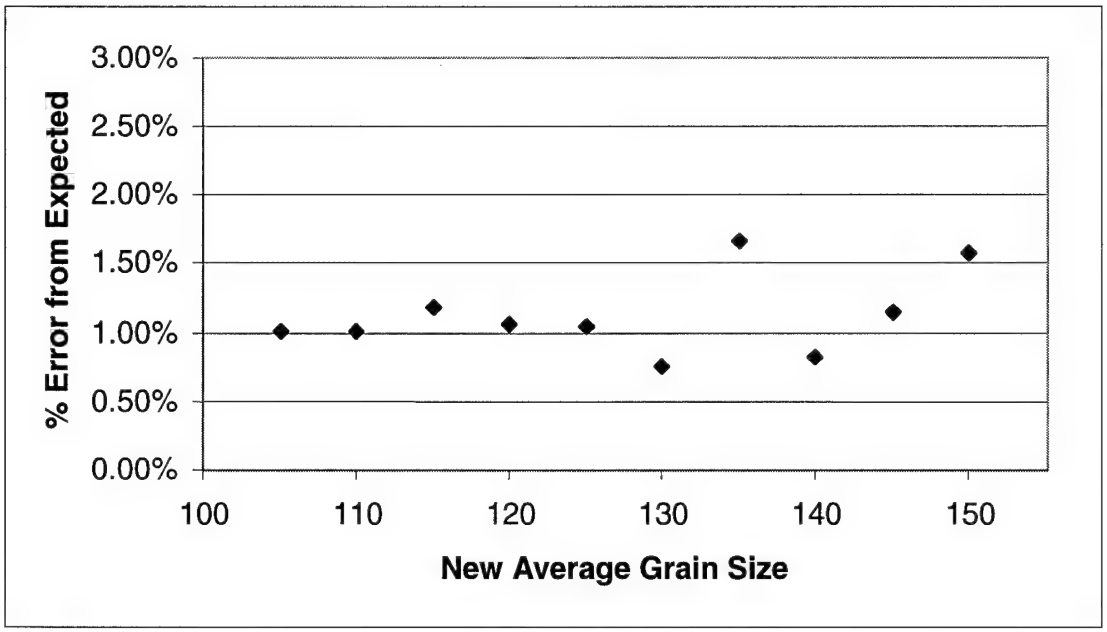


Figure 3-23. Average Grain Size Error as a Function of Expected New Average Grain Size.

3.5.3 Grain Size Distribution Changes Due to Grain Growth

The tests performed in this section show the effects of grain growth on grain size distribution. The first series of tests show relationships between the grain size distribution curve as the average grain size is increased. The next set of tests normalizes the grain size distribution curves to show that the general shape of the distribution curve is independent of grain growth.

This first series of tests display the shifting of the grain size distribution curve as a result of grain growth. For this set of tests, an initial microstructure of 100 grains with an average grain size of 100 microns is generated. The grain growth algorithm is run for new average grain sizes of 110, 120, 130, and 150 microns. The grain size distribution

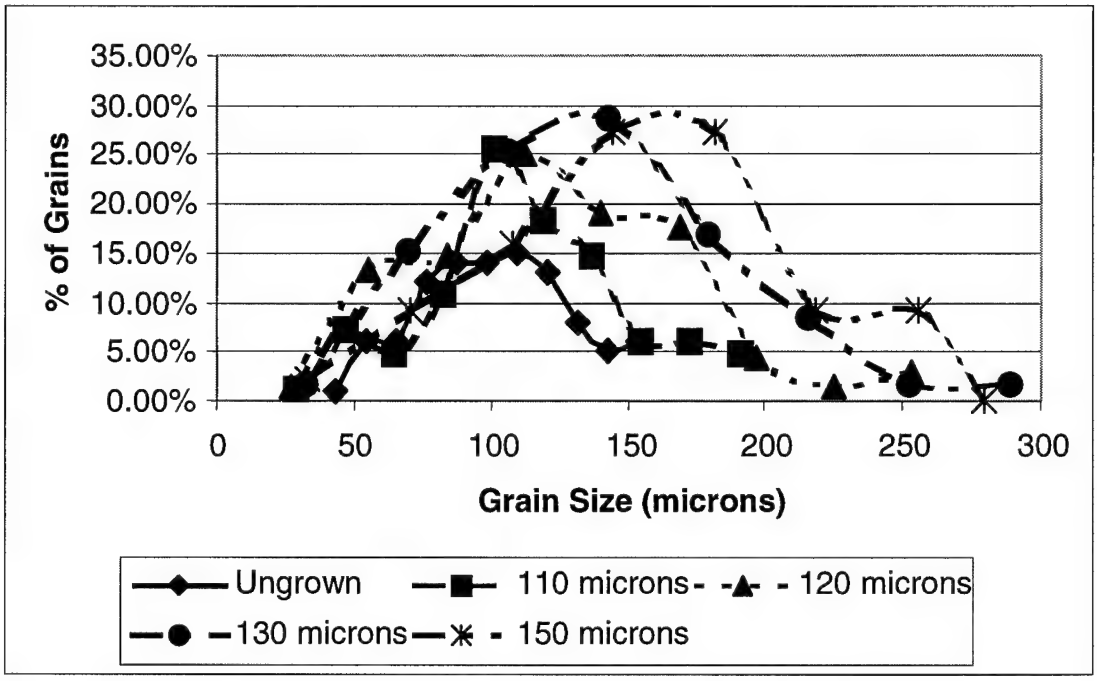


Figure 3-24. Grain Size Distributions for Different Grain Growths of a 100 Micron Initial Grain.

for the new microstructures are compared with the grain size distribution of the initial microstructure and displayed in Figure 3-24.

The next set of tests compare a normalized set of grain distribution curves for different new average grain sizes. For this series of tests, an initial microstructure with an average grain size of 100 microns and 100 grains is again generated. The grain growth algorithm is performed for new average grain sizes of 110, 120, 130, and 150 microns. In order to normalize each grain distribution, the grain size for each bin is divided by the value of the bin containing the average grain size for that distribution. The normalized results from the tests are compared with the normalized distribution of the original and displayed in Figure 3-25.

3.5.4 Visual Comparison of Grain Growth Simulations

This final section of the chapter visually compares the results of grain growth performed by the simulation to grain growth performed by another simulation from the

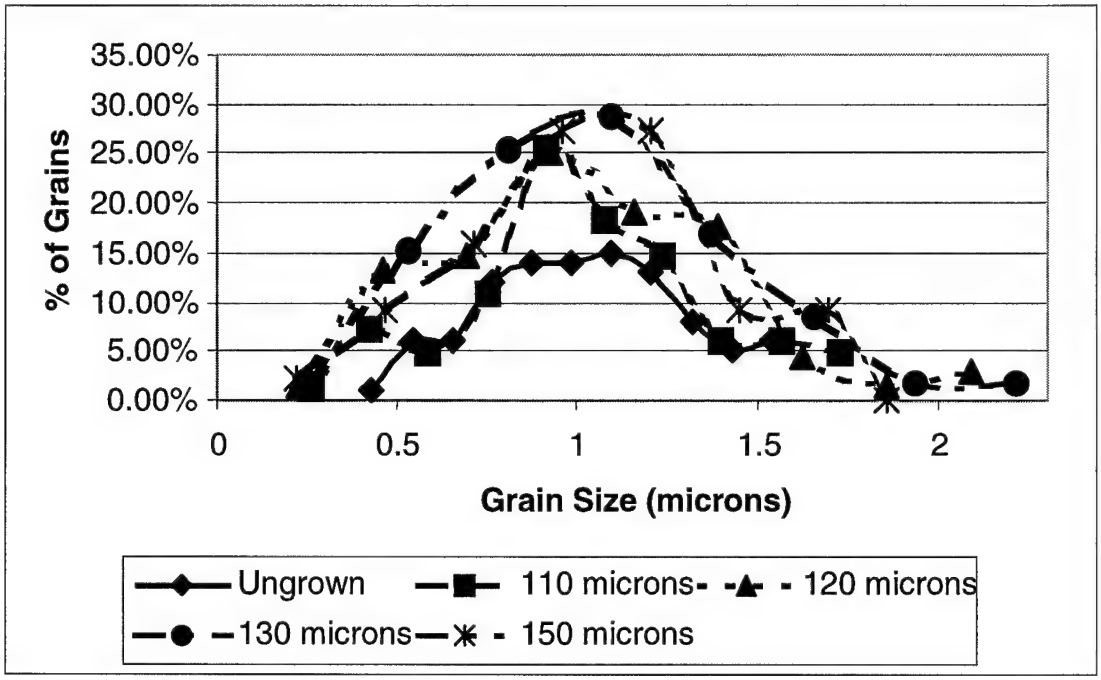


Figure 3-25. Normalized Grain Size Distributions for Different Grain Growths of a 100 Micron Initial Grain.

literature. For this comparison, a microstructure with 100 grains is generated with an average grain size of 100 microns. The microstructure is grown to an average grain size of 105 microns and then to an average grain size of 110 microns. The simulated microstructures are compared in Figure 3-26 to microstructures generated by Anderson et. al (1986).

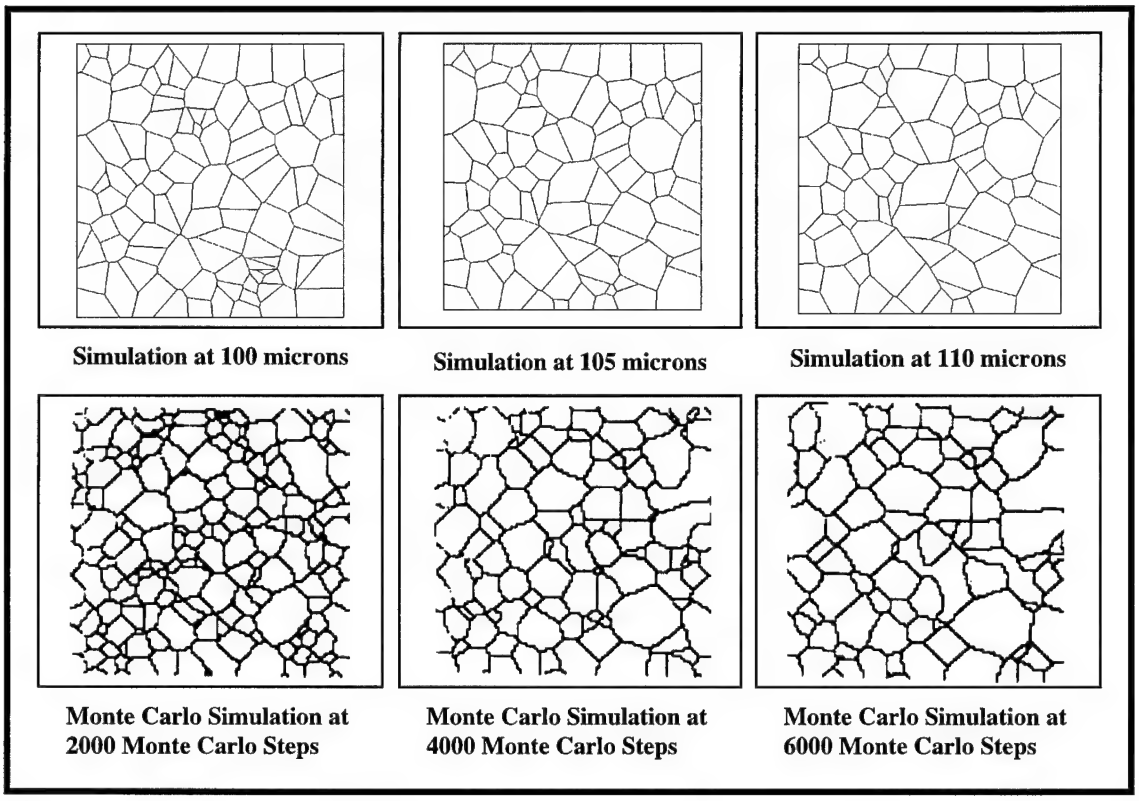


Figure 3-26. Comparisons of Simulated Grain Growth for Different Simulations. Monte Carlo Simulations Taken from Anderson et. al. (1986).

CHAPTER 4

CONCLUSION

4.1 Introduction

This chapter concludes the thesis by presenting an appraisal of the results, an appraisal of the design, areas of future development and improvement, and finally, a summary of this thesis. The appraisal of results section discusses the results found in Chapter 3. The next section on the appraisal of the design approach discusses the methodology used in the design of the simulation. After the appraisal of the design, a section on future areas of development is presented, describing improvements and further work that can be explored based on the results of this research. The final section summarizes and concludes this paper.

4.2 Appraisal of Simulation Results

This section appraises the results presented in Chapter 3 on the performance of the four algorithms. The first subsection discusses the results from the tests performed on the initial microstructure algorithm. Results from the grain deformation tests are appraised in the second-subsection. The next subsection appraises the results from the grain recrystallization tests. The results from the grain growth algorithm are discussed in the final subsection.

4.2.1 Appraisal of Initial Microstructure Algorithm Tests

This subsection appraises the tests performed in Section 3.2 of this thesis. First, an appraisal is made of the time-to-solution tests. Then, the grain distribution tests are examined. The grain distribution comparisons with reference tests are discussed. Finally, an analysis of the visual comparison test ends this subsection. The significance of each test is fully explored in this subsection.

The time-to-solution tests performed on this algorithm show several qualities of the algorithm's performance. The most obvious quality is that the time to solution for the algorithm is solely dependent on the number of grains, as is shown in Figure 3-1. In fact, the time tends to be a function of the cube of the number of grains. Another quality is that the average grain size has very little effect on the time to solution, as shown in Figure 3-2. Thus, the conclusion for the time to solution for the initial microstructure algorithm is that it is dependent on the cube of the number of grains and independent of the average grain size.

The grain size distribution comparisons show the robustness of the solution with respect to the number of grains and to the average grain size. The grain size distribution is shown to be highly consistent within the range of 50 to 500 grains (Figure 3-3). When the grain size distribution is normalized with respect to the average grain size, and distributions for different grain sizes are compared, the shapes of the distribution curves are very similar for grain sizes ranging from 10 to 100 microns (Figure 3-4). Based on the grain size distribution comparisons, the algorithm is very robust for providing consistent distributions of grains for a variety of numbers of grains and grain sizes.

The grain size distribution comparisons with reference microstructures provide validation that this algorithm is producing microstructures that reasonably represent actual microstructures or microstructures simulated by proven methods. The comparison between the grain size distributions between the simulation and an actual microstructure (Figure 3-5) show that although the distributions have the same characteristic shape, the distribution of the actual microstructure has greater variation (it is flatter and more spread out over the range of grain sizes). This fact is expected since the simulated microstructure represents an idealized microstructure and not a real microstructure. The comparison between the size distribution generated by this simulation and another simulation (Figures 3-6 and 3-7) show quite similar distributions. These comparisons show that although the grain size distribution for the simulation does not exactly match the actual microstructure, it has a very similar shape and does match grain size distributions for the other proven simulation. Therefore, this simulation gives a grain size distribution that provides a reasonable approximation of real microstructures and the results are similar to other proven simulations.

The visual comparison is actually the most important comparison even though it is qualitative rather than quantitative. This importance stems from the fact that the object of this thesis is to develop a simulation, which generates microstructures that are visually close to actual microstructures. Figure 3-8 shows that the microstructure generated with this simulation has a very similar appearance to microstructures generated by other proven simulations. This similarity leads to the conclusion that this simulation produces microstructures that represent actual microstructures in a manner that appears equal with other proven programs.

The initial microstructure algorithm tests show several important facts about the simulation. First, the amount of time that the simulation requires to produce a microstructure depends greatly on the number of grains and has very little dependence on the average grain size. The simulation is robust and generates consistent grain size distributions for virtually any number of grains at any grain size. Finally, these tests show that the simulation produces approximations to actual microstructures that are at least as good as other proven programs.

4.2.2 Appraisal of Grain Deformation Algorithm Tests

The tests performed in Section 3.3 are appraised in this subsection. The results of the grain deformation algorithm's time-to-solution test are analyzed first. After the time-to-solution results are analyzed, the difference between the actual results and the expected results are examined. Finally, an analysis of the visual comparison is made between a microstructure deformed through the simulation and samples of deformation from actual microstructures.

The time-to-solution results presented in Figure 3-9 show the time the algorithm takes to deform the initial microstructure. The most obvious feature of the graph is the results for some of the tests show a value of zero seconds, and none of the tests took longer than six milliseconds for the algorithm to perform. The reason for the zero seconds results is simply that the algorithm performs faster than the resolution of the timer function used. The final conclusion based on the results is that the time for this algorithm to deform the microstructure is almost negligible when compared to the other algorithms.

The difference between the actual deformation and the expected deformation as entered by the user defines the reliability of the algorithm. The graph in Figure 3-10 shows the difference between the actual and expected percent deformations as a function of the percent strain for several numbers of grains. As the graph shows, there is no significant difference between the actual and expected deformation for percent strains up to 60 percent. Strains above 60 percent only produce less than a 0.01 percent difference, which is not significant because the error is independent of the number of grains in the microstructure. These tests show that the difference between the actual and expected results are virtually negligible, which implies that the algorithm is very reliable.

The visual comparison gives a qualitative analysis of how the simulated deformed microstructure compares with an actual deformed microstructure. Figures 3-11 and 3-12 show results from deformations of the simulation and an actual microstructure. The deformations range from 10 to 60 percent strain. For each sample, the characteristic grain elongation associated with grain deformation is present in both the simulated microstructure and the actual microstructure. Additionally, the visual similarities are obvious between the simulated and actual microstructures. The conclusion from the visual comparison is that the simulated deformed microstructure exhibits the same characteristics as the actual deformed microstructures.

These tests are performed on the grain deformation algorithm in order to determine the reliablity and accuracy of the algorithm. The tests performed to determine the time to solution show that the algorithm takes an insignificant amount of time to produce results. The difference between the actual and expected deformations are shown to be very small proving that the algorithm is reliable. The visual comparison with actual

deformed microstructures qualitatively show that the algorithm produces a deformed microstructure that represents actual microstructures.

4.2.3 Appraisal of Grain Recrystallization Tests

This subsection analyzes the results for the grain recrystallization algorithm presented in Section 3.4. An analysis of the algorithm's time to solution is presented first. The comparison between the expected and actual recrystallized grain size and percent recrystallization is presented next. The effect of recrystallized grain size and percent recrystallization on grain size distribution are then presented. Lastly, a qualitative comparison is made between simulated and actual recrystallized microstructures.

The tests performed to determine the time to solution for the grain recrystallization algorithm show that the time generally depends on both the recrystallized grain size and the percent recrystallization. According to Figure 3-13, the time to solution increases exponentially as the percent recrystallization increases. Figure 3-14 shows that the time to solution decreases on the order of the inverse of the square of the recrystallized grain size. These tests show that the time to solution increases as the percent recrystallization increases, but decreases as the recrystallized grain size increases.

The comparison between the expected and actual recrystallized grain size and percent recrystallization show the accuracy of the simulation as compared with actual microstructures. Figure 3-15 shows that the error between the actual and expected grain size decreases as the percent recrystallization increases. A reduction in error for percent recrystallization as the percent recrystallization increases is also shown in Figure 3-16.

The reduction in error for percent recrystallization and recrystallized grain size is due to the fact that as the percent recrystallization increases, the number of recrystallized grains increases. This increase in grains causes the actual average recrystallized grain size and percent recrystallization to approach the expected values. The graphs in Figures 3-17 and 3-18 show that the error in the recrystallized average grain size and the percent recrystallization decrease as the recrystallized grain size approaches half of the initial average grain size, an error due to the fact that as the average recrystallized grain size deviates from half the value of the initial grain size, the microstructure has to correct for grain size error caused by the Voronoi algorithm. This correction algorithm increases the error that enters into the algorithm. The conclusions drawn from these tests are that as the percent recrystallization increases and the recrystallized grain size approaches about half of the initial average grain size, the recrystallization algorithm becomes more accurate.

An analysis of the grain size distributions shows that they are affected by recrystallized grain size and percent recrystallization. Figure 3-19 shows that as the recrystallized average grain size decreases, the distribution shifts to the left and a bimodal distribution for grain size typical of recrystallized microstructures appears. Unlike the recrystallized average grain size, the grain size distribution curve shifts to the left as the percent recrystallization increases (Figure 3-20). This shift is expected for both the decrease in the recrystallized grain size and the increase in the percent recrystallization, because for both cases, the number of smaller, recrystallized grains increases.

The final test performed with the recrystallization algorithm is a qualitative comparison between simulated recrystallized microstructures and actual recrystallized

microstructures. Figure 3-21 shows simulated and actual microstructure. Both the simulated microstructures and the actual microstructure show the typical division of large,coarse grains and the small, fine grains. This test shows that the simulation produces recrystallized microstructures that have a similar appearance to actual recrystallized microstructures.

The conclusions that can be drawn about the recrystallization algorithm involve the time to solution, reliability, and visual quality. The time to solution decreases as the recrystallized grain size increases, and increases as the percent recrystallization increases. Also, as the percent recrystallization increases and the average recrystallized grain size approaches half of the initial grain size, the difference between the expected and actual decreases. Finally, the algorithm produces recrystallized microstructures that are visually similar to actual microstructures.

4.2.4 Appraisal of Grain GrowthTests

The appraisal of the grain growth tests in Section 3.5 is presented in this subsection. First, an analysis of the algorithm's time to solution is presented. Following this analysis, the expected new average grain size as compared with the actual new average grain size is explored. The effect of grain growth on grain size distribution is compared for different new grain sizes. The last topic in this subsection is an analysis of visual comparisons between the simulation's grain growth algorithm and a Monte Carlo grain growth simulation found in the work of Anderson et al. (1986).

The results of the time-to-solution test determine the performance of the grain growth algorithm. Figure 3-22 shows that the time to solution increases as the new

average grain size increases. The increase is expected because as the new average grain size increases, there is a greater number of grains to be removed. The more grains that need to be removed, the longer the amount of time required by the algorithm to generate the microstructure. However, the time to solution for these tests were less than 2.5 seconds, which is not an unreasonable wait time.

The difference between the expected new average grain size and actual new average grain size is presented in Figure 3-23. This test shows that the new average grain size has very little effect on the difference between actual and expected new average grain size, and the differences for the tests were less than two percent. The results show that the simulation gives consistent microstructures that are produced by the grain growth process.

The grain size distribution tests show how they change due to grain growth. Figure 3-24 shows that the grain size distribution shifts to the right (larger grain sizes) as a result of grain growth. The normalized grain size distribution in Figure 3-25 shows that the shape of the curve changes little as the grain size increases. These tests show that the grain growth shifts the grain size distribution towards larger grain sizes while maintaining the shape of the distribution curve.

The last test performed on the grain growth algorithm is a visual comparison between this program's grain growth algorithm and a Monte Carlo method of grain growth. Figure 3-26 shows this comparison. For this program's simulated grain growth algorithm, the figure shows the grain growth and reduction of the number of grains as the new grain size increases. The Monte Carlo simulation shows the same grain growth and

reduction in number of grains. The results show that this simulation produces grain growth similar to growth produced in other proven simulations.

Finally, the grain growth algorithm analysis appraises the time to solution, the error between the actual and expected new average grain sizes, the grain size distribution, and a visual comparison between this simulation and another proven simulation. The test for the time to solution shows that this algorithm completes its operation within a reasonable time. The small error between the actual average grain growth and the expected grain growth show that the algorithm is consistent. The grain size distribution change due to grain growth shows that the algorithm produces an expected distribution shift. Finally, the visual comparison shows that the algorithm produces a simulation similar to simulations produced by proven techniques.

4.3 Appraisal of the Design of the Simulation

This section on the appraisal of the design is divided into three subsections. The first subsection analyzes the object-oriented programming methodology used to design the simulation. The second subsection discusses the method used to represent the microstructure. The last subsection analyzes the organization and design of the four algorithms.

4.3.1 Appraisal of Object-Oriented Programming Methodology

The object-oriented design methodology is used extensively in the design of this simulation. The program was designed to be modular following the object-oriented philosophy. This modularity allows great flexibility in the design, implementation, testing, and future improvement of the simulation. Object-oriented methods allow the

program to be designed in a highly abstract manner. Instead of focusing on the minor nuances of the code, the overall simulation was first designed in a conceptual manner by dividing the simulation into objects with attributes and methods associated with each object. By breaking the overall architecture into separate objects, the code was designed to be very modular which allowed it to be written one object at a time. This modularity also permitted each object to be tested independent of each other, which saved in comparison to the overall time it would have taken to write the code under another design methodology. A final advantage to the object-oriented approach is the in the ability to upgrade and improve this software. Each object can be improved independently without having to make major changes to the rest of the program. Also, new objects can be added to the program quickly and easily to improve functionality without having to redesign the whole program.

4.3.2 Appraisal of Microstructure Representation

The use of object-oriented methods allowed the microstructure to be represented in an abstract manner. The design of the microstructure allows it to be modular as it can be moved to other programs needing to represent microstructure information. The information in the microstructure consists of grain objects, triple point objects, and functions which manipulate the microstructure. The representation of the microstructure allows it to be changed easily should other microstructure information be needed.

The microstructure is represented as an object that can be implemented by another program. In the case of this thesis, a simulation was written using Microsoft Foundation Classes to allow the microstructure object to be implemented and displayed on the

computer screen for purposes of testing. However, the object-oriented design of the microstructure object allows the code to be reused in any other program designed for Windows 95/NT.

The microstructure object contains lists of other objects such as the grain object list and the triple point object list that define the microstructure. The grain objects and triple point objects define the overall microstructure and each object contains the information that defines the object as well as any functions designed to act on that object. Also, the microstructure object contains functions associated with the manipulation of the microstructure. These functions include the initial microstructure algorithm, the grain deformation algorithm, the grain recrystallization algorithm, the grain growth algorithm, and other minor manipulation functions associated with the microstructure.

The object-oriented representation of the microstructure allows it to be updated and changed rapidly and easily. For example, information such as grain orientation and phase can be added to the grain object with minor modifications to the rest of the microstructure object. Minor modifications would have to be made to each of the process algorithms, but the modularity allows the modifications to be implemented and tested individually and quickly.

Overall, the object-oriented microstructure representation is very beneficial. The representation allows the microstructure to be defined logically. The microstructure object can be implemented into other programs easily. Finally, future improvements and modifications to the microstructure object can be designed, implemented, and tested easily and quickly.

4.3.3 Appraisal of Algorithm Organization and Design

Following the object-oriented philosophy, the processes that act on microstructures during manufacturing processes are represented as separate functions of the microstructure object. They are divided into the algorithm that creates the initial microstructure, the grain deformation algorithm, the grain recrystallization algorithm, and the grain growth algorithm. The object-oriented method allows these algorithms to be designed, implemented, and tested individually, which speeds coding time and improves the quality of the code.

The initial microstructure algorithm uses a Voronoi polygon method to create the initial microstructure. This method is widely used to represent microstructures and is fairly easy to implement. The Voronoi algorithm used in this program is known as the "brute force" version. For the purposes of this prototype, the algorithm functions well enough to prove the concept, but for a production version, the algorithm's speed would need to be improved. Other Voronoi algorithms exist, but these algorithms were designed for a Unix platform. A search on the Internet through computational geometry web sites during the literature review produced many algorithms for Unix but none for a Windows platform. However, the Unix algorithms could be rewritten for a Windows platform.

The grain deformation algorithm was designed to show primarily the elongation effects that appear in deformed grains. Because little quantification of grain deformation exists in the literature and the simulation is only required to give a general representation, simple scaling algorithms were used to deform the grain. The scaling algorithms allowed the simulation to represent characteristics of grain deformation in a general manner,

which, as shown in the results, give a very good approximation of actual grain deformation.

The grain recrystallization algorithm was designed to visually represent the actual recrystallization process. Although, the literature contains much thought as to the location of recrystallization sites, recrystallization along grain boundaries is widely accepted, so the algorithm focuses on the recrystallization along grain boundaries. However, the modularity of the simulation allows this algorithm to be changed to reflect improved knowledge of the recrystallization process. Since this algorithm, like the initial grain algorithm, utilizes a Voronoi polygon method, alternative Voronoi methods could improve this algorithm's speed. Also, either decreasing the threshold used to correct grain size anomalies due to the Voronoi algorithm or using a weighted Voronoi method would improve the results of the algorithm. However, decreasing the threshold would also decrease the speed of the algorithm. A weighted Voronoi algorithm, using weighted values for each polygon to produce a variety of distributions, would be faster and more accurate, but is difficult to implement.

The grain growth algorithm works generally well, but during testing, the program crashed several times. The crashes were due to the geometric complexity of the grain growth process. Because of the randomness of the program, the geometric anomalies that crashed the grain growth algorithm did not appear all of the time. Therefore, the problems with geometric complexity remain in the prototype code produced for this study. The instability of the algorithm does not affect the ability to use and test the algorithm, but it should be addressed in a production version of the simulation.

The method of dividing the program into four separate algorithms worked very well. This division allowed each algorithm to be designed and tested separately. This improved the quality of the overall program and shortened the design and implementation time. Finally, the organization of the algorithm allows any modifications to be performed easily and quickly.

4.4 Future Areas of Development and Improvement

Future areas of work in relation to this thesis are suggested and discussed in this section. The first area of work described in the first subsection is adding simulation of multi-phase metals and precipitates to the functionality of the program. The second subsection discusses the area of improving the efficiency of the program. The last subsection describes the area of integrating the simulation with other programs.

4.4.1 Adding the Capability to Simulate Multi-phase Metals and Precipitates in a Microstructure

One of the future areas of improvements that can be made to this simulation is the capability of simulating multi-phase metals and precipitate laden metal. Since the prototype is designed to prove the concept only, it simplifies the microstructure representation into a single-phase metal to provide a well bounded problem. However, many common metals, such as steel, are multi-phase and also contain precipitates that affect the different metal processes. Information could easily be added to the microstructure object and grain objects to identify different phases and the addition of precipitates. The other algorithms would have to be redesigned to process the new information, but the modularity of the program should make such modifications fairly

easy to implement. The result would be a program that more accurately simulates real microstructures.

4.4.2 Improving Efficiency of Simulation

Since this simulation was designed to be a prototype to prove the concept, little emphasis was placed on the efficiency of this simulation. The greatest improvement to the simulation would be optimization of the Voronoi algorithm. The Voronoi method used in this program is a very inefficient algorithm. As previously mentioned other Voronoi algorithms have been designed for Unix platforms and would have to be rewritten for a Windows platform. Another improvement to the efficiency would be the addition of a weighted Voronoi method to the recrystallization algorithm. The weighted Voronoi would require some time to implement but would improve the speed and accuracy of the algorithm. Improvements to efficiency are necessary in order to develop a simulation from the prototype to a production model.

4.4.3 Integrating Simulation into Other Programs

The next stage of development for this simulation is to integrate a production model version of this prototype into other programs. In order to reach that stage, several modifications have to be made. The first and most important change, previously mentioned, is the improvement of the speed of the simulation. Also, it is important to further debug the grain growth algorithm to remove that algorithm's instability for a production model of the code. Finally, the prototype algorithm needs to be converted into library files. These library files could be of either the dynamically linked or static

type. Converting the code into a library file is a simple process that compilers such as Visual C++ can perform and makes it easier to implement into other programs.

4.5 Concluding Summary

The purpose of this thesis is to develop a computer simulation that can visually show microstructures and the effects of metals process, such as grain deformation, recrystallization, and growth. The simulation is designed to able to integrate with a manufacturing process simulation being developed by the Materials and Manufacturing Directorate of the Air Force Research Laboratory at Wright-Patterson Air Force Base in Ohio or any other metals related software. An object-oriented approach is used to design the software which satisfies the purpose of this thesis. The simulation is then tested to determine how well it performs and simulates actual microstructures. According to the results of the tests, the simulation, although crude and inaccurate, is able to approximate the visual representation of microstructures as they undergo metal processes. More work needs to be done with this prototype in order to develop it into a useful production-level simulation, but this research effort has proven capabilities for an initial Windows 95/NT based system for visual simulation of the microstructure of metals.

REFERENCES

1. American Society for Metals, 1985, "Metallography and Microstructures," *Metals Handbook*, 9th Ed., American Society for Metals, Metals Park, OH, Vol. 9.
2. American Society for Metals, 1973, "Metallography, Structures and Phase Diagrams," *Metals Handbook*, 8th Ed., American Society for Metals, Metals Park, OH, Vol. 8, pp. 37-47.
3. Anderson, M. P., Grest, G. S., and Srolovitz, D. J., 1986, "Microstructural Dynamic Study of Grain Growth," *Computer-Based Microscopic Description of the Structure and Properties of Materials*, Pittsburgh, PA, pp. 225-231.
4. ASTM, 1989, "Metals -- Mechanical Testing; Elevated and Low-Temperature Tests; Metallography," *Annual Book of ASTM Standards*, ASTM, Philadelphia, PA, Vol 03.01, pp. 284-309.
5. Burke, J.E., and Turnbull, D., 1952, "Recrystallization and Grain Growth," *Progress in Metal Physics*, Interscience Publishers, Inc., New York, pp. 220-292.
6. Cerezo, A., Hetherington, M. G., Hyde, J. M., Miller, M. K., Smith, G. D. W., and Underkoffler, J. S., 1992, "Visualisation of Three-Dimensional Microstructures," *Surface Science*, Elsevier Science Publishers, pp. 471-480.
7. Devadas, C., Samarasekera, I.V., and Hawbolt, E.B., 1991, "The Thermal and Metallurgical State of Steel Strip During Hot Rolling: Part I. Characterization of Heat Transfer," *Metallurgical Transactions*, Vol. 22A, pp. 307-319.

8. Devadas, C., Samarasekera, I.V., and Hawbolt, E.B., 1991, "The Thermal and Metallurgical State of Steel Strip During Hot Rolling: Part III. Microstructural Evolution," *Metallurgical Transactions*, Vol. 22A, pp. 335-349.

9. Enomoto, M., Salto, Y., Fuchizaki, K., Nagal, T., and Kawasaki, K., 1993, "Evolution of 2-D Polycrystalline Grain Structure Simulated by Vertex Model," *Proceedings, International Conference on Computer-Assisted Materials Design and Process Simulation: COMMP '93*, Tokyo, pp. 380-385.

10. Fredriksson, H., 1990, "Mechanism of Grain Growth in Metals," *Materials Science and Technology*, The Institute of Metals, pp. 811-817.

11. Frost, H. J., and Thompson, C. V., 1986, "Microstructural Evolution in Thin Films," *Computer Simulation of Microstructural Evolution*, The Mettallurgical Society, Warrendale, PA, pp. 33-47.

12. Furu, T., Marthinsen, K., and Nes, E., 1990, "Modeling Recrystallisation," *Materials Science and Technology*, The Institute of Metals, pp. 1093-1102.

13. Grest S., Anderson, M. P., and Srolovitz, D. J., 1986, "Computer Simulation of Microstructural Dynamics," *Computer Simulation of Microstructural Evolution*, The Mettallurgical Society, Warrendale, PA, pp. 21-32.

14. Krakow, W., 1995, "Modelling of Two-Dimensional Grain Growth and Crystallization," *Proceedings, Materials Research Society Symposia*, Materials Research Society, New York, pp. 25-30.

15. Malas, J. C., and Frazer, W.G., 1997, "Perspectives on New Design Tools for Material Processes," *JOM*, The Minerals, Metals & Materials Society, New York, Sept., pp. 32-33.

16. Malas, J. C., Medina, E. A., and Dubrosky, B. M., 1997, "Interactive Simulation System for Design of Multi-Stage Material Processes," *Proceedings of the 1997 International CIRP Design Seminar on Multimedia Technologies for Collaborative Design and Manufacturing*, University of Southern California, Los Angeles, Oct. 8-10, pp. 267-275.

17. Porter, D. and Easterling, K. E., 1992, *Phase Transformations in Metals and Alloys*, Chapman and Hall, New York.

18. Rappaz, M., Gandin, C. A., Jacot, A., and Charbon, C., 1995, "Modeling of Microstructure Formation," *Modeling of Casting, Welding, and Advanced Solidification Processes VII*, The Minerals, Metals, and Materials Society, pp. 501-516.

19. Roberts, W., 1986, "Microstructure Evolution and Flow Stress During Hot Working," *Strength of Metals and Alloys (ICSMA 7)*, Pergamon Press, Oxford, Vol. 3, pp. 1859-1891.

20. Sakai, T., and Jonas, J. J., 1984, "Dynamic Recrystallization: Mechanical and Microstructural Considerations," *Acta Metallurgica*, Pergamon Press, pp. 189-209.

21. Takayama, Y., Tozawa, T., Kato, H., and Furushiro, N., 1996, "Computer Simulation of Grain Growth Induced by Superplastic Deformation," *Materials Science Forum*, Transtec Publications, Aedermannsdorf, Switzerland, pp. 319-324.

22. Vander Voort, G. F., 1984, "Grain Size Measurement," *Practical Applications of Quantitative Metallography*, American Society for Testing and Materials, Ann Arbor, MI, pp. 85-131.

23. Vander Voort, G. F., 1984, *Metallography Principles and Practice*, McGraw-Hill Book Company, New York.